



โครงการวิทยาการคอมพิวเตอร์

“เว็บแอปพลิเคชันร้านค้าออนไลน์ ไฮบลัด”

(HiBLOOD Shop Web Application)

ผู้จัดทำ

นางสาวสุชาดา	มาลัยเวช	รหัสนิสิต 6010450772
นางสาวฉันทิกา	บำรุง	รหัสนิสิต 6010451027

เสนอ

ผศ.ดร.อุษา สัมมาพันธ์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเกษตรศาสตร์ ภาควิชา ศึกษาศาสตร์ ปีการศึกษา 2563

โครงการด้านวิทยาการคอมพิวเตอร์

หัวข้อ เว็บแอปพลิเคชันร้านค้าออนไลน์ ไฮบลัด

(HiBLOOD Shop Web Application)

โดย นางสาวสุชาดา มาลัยเวช รหัสனிสิต 6010450772

นางสาวฉันทิกา บำรุง รหัสனிสิต 6010451027

อาจารย์ที่ปรึกษา _____

(ผศ.ดร.อุษา สัมมาพันธ์)

_____/_____/_____

อาจารย์ที่ปรึกษาร่วม _____

(ผศ.ศิริกร จันทน์นวล)

_____/_____/_____

อาจารย์ที่ปรึกษาร่วม _____

(ผศ.พบสิทธิ์ กมลเวช)

_____/_____/_____

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์ _____

(ผศ.ดร.ผกาเกษ วัตุยา)

_____/_____/_____

ชื่อหัวข้อโครงการ HiBLOOD Shop Web Application

โดย นางสาวสุชาดา มาลัยเวช รหัสนิสิต 6010450772

นางสาวฉันทิกา บำรุง รหัสนิสิต 6010451027

คำสำคัญ (keyword) Web Application, แลกคะแนน

อาจารย์ที่ปรึกษา _____

(ผศ.ดร.อุษา สัมมาพันธ์)

บทคัดย่อ

ในปัจจุบัน “โลหิต” หรือ “เลือด” ยังไม่สามารถคิดค้นสิ่งใดมาทดแทนได้จึงจำเป็นต้องการมีผู้บริจาคเลือด “การบริจาคเลือด” คือการสละโลหิตส่วนเกินที่ร่างกายยังไม่จำเป็นต้องใช้ให้กับผู้ป่วยซึ่งไม่เป็นอันตรายต่อผู้บริจาคเลือด โดยสามารถบริจาคโลหิตได้ทุก 3 เดือน เมื่อบริจาคเลือดแล้วไขกระดูกจะเป็นส่วนสำคัญในการสร้างเม็ดเลือดขึ้นมาทดแทน ทำให้มีปริมาณเลือดในร่างกายเท่า เดิมหากไม่ได้บริจาค ร่างกายจะขับเม็ดเลือดที่สลายตัวเพราะหมดอายุ⁽¹⁾ ในทุกวันมีผู้ต้องการเลือดเพื่อรักษาและช่วยชีวิตผู้ป่วยเป็นจำนวนมากทำให้เกิดการขาดแคลนเลือดที่ศูนย์บริการโลหิตแห่งชาติสภากาชาดไทยบ่อยครั้งรวมถึงโรงพยาบาลอื่นๆอีกด้วย รวมถึงเกิดสถานการณ์โรคระบาดจากไวรัสโคโรนา (Covid-19) เกิดเหตุการณ์ลดลงของผู้บริจาคเป็นจำนวนมากถึงเข้าขั้นวิกฤต ผู้จัดทำจึงเกิดการจัดทำเว็บแอปพลิเคชันเพื่อให้ผู้บริจาคเลือดนั้นสามารถนำจำนวนครั้งการบริจาคเปลี่ยนเป็นแต้มคะแนนแล้ว นำแต้มคะแนนนั้นมาลดราคาสິงของในร้านค้าของเรา เราได้อ้างอิงจากงานวิจัย คุณอุสา สุทธิสาคร, คุณสิทธิพงศ์ วัฒนานนท์สกุลและคุณดำรง เชี่ยวศิลป์ ได้ทำการวิจัยพฤติกรรมการบริจาคโลหิตของผู้ที่มาบริจาคโลหิต ณ ศูนย์บริการ โลหิตแห่งชาติ สภากาชาดไทย⁽²⁾ ส่วนใหญ่มาจากการเอื้อเฟื้อและแรงจูงใจที่กลับมาคือความเอื้อเฟื้อเช่นกัน แรงจูงใจที่ทำให้อยากมาบริจาคเลือดก็คือต้องการช่วยเหลือเพื่อนมนุษย์ เว็บแอปพลิเคชันเราเปรียบเหมือนแทนการขอบคุณและให้กำลังใจกับผู้บริจาคเลือดที่มาบริจาคเลือดอย่างต่อเนื่องหลายๆครั้งและเป็นการกระตุ้นทางอ้อมให้เกิดเป้าหมายนอกเหนือจากการเอื้อเฟื้อให้ผู้บริจาคเลือดที่เคยลงใช้บริการบริจาคเลือดแล้วนั้นอาจจะหยุดบริจาคไปหรือผู้ไม่เคยบริจาคสามารถทำการบริจาคอย่างต่อเนื่องจนครบเป้าหมายของแต้มคะแนนนั้น

เว็บแอปพลิเคชันเราจะกำหนดการบริจาคทุกๆ 1 ครั้ง เท่ากับ 450 คะแนนและแต้มคะแนน 1 คะแนน เท่ากับ 1 บาท โดยแต้มคะแนนไม่มีวันหมดอายุเพื่อให้ผู้บริจาคค่อยๆสะสมแต้มคะแนนจนสามารถลดราคาสินค้า นั้นจนสามารถลดราคาสินค้าเหลือเพียง 0 บาทจนเหมือนได้ฟรีนั่นเอง หากแต้มคะแนนสะสมไม่ถึงก็สามารถจ่าย ส่วนต่างที่ไม่ได้ลดราคา สามารถดำเนินการชำระเงินผ่านบัตรเครดิต/เครดิต ก็ทำได้อย่างปลอดภัย

คำนำ

การจัดทำเว็บแอปพลิเคชัน Hiblood นี้มีจุดประสงค์เพื่อให้เกิดการกระตุ้นทางอารมณ์และอยากขอบคุณให้กับผู้บริจาคเลือดที่มาบริจาคเลือดเพิ่มมากขึ้นและยังเกิดการบริจาคครั้งต่อไปโดยใช้จากการสะสมแต้มคะแนน แล้วนำแต้มคะแนนนั้นมาเป็นส่วนลดให้กับสิ่งของ โดยเอกสารฉบับนี้เป็นส่วนหนึ่งของวิชา 01418499 Computer Science Project ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ภาควิชาการศึกษาศาสตร์ 2563

คณะผู้จัดทำหวังว่าเอกสารฉบับนี้ จะเป็นประโยชน์แก่ผู้ที่สนใจการทำเว็บแอปพลิเคชัน ได้นำไปใช้ในการพัฒนาในอนาคตต่อไป หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย

นางสาวสุชาดา วัลย์เวช รหัสนิสิต 6010450772

นางสาวฉันทิกา บำรุง รหัสนิสิต 6010451027

ผู้จัดทำ

กิตติกรรมประกาศ

ในการจัดทำโครงงานปัญหาวิทยาการคอมพิวเตอร์ เรื่อง “เว็บแอปพลิเคชันไฮบลัด” (HiBLOODShop Web Application) รายงานวิจัยนี้เป็นส่วนหนึ่งของวิชา 01418499 Computer Science Project ภาควิชา วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ภาควิชา ปีการศึกษา 2563 คณะผู้จัดทำได้ ทำโครงงานปัญหาพิเศษประสบความสำเร็จได้ความอนุเคราะห์และได้คำแนะนำจากอาจารย์จึงขอขอบพระคุณ อาจารย์ทุกท่านต่อไปนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ.ดร.อุษา สัมมาพันธ์ อาจารย์ที่ปรึกษาโครงงาน ผู้ให้คำปรึกษาและ แนะนำแนวทางการแก้ไขในการดำเนินงานต่างๆคณะผู้จัดทำขอกราบขอบพระคุณอาจารย์เป็นอย่างสูง

ขอขอบพระคุณอาจารย์ ผศ.ศิริกร จันทร์นวล และ ผศ.พบสิทธิ์ กมลเวช อาจารย์ที่ปรึกษาร่วม ผู้ให้ คำปรึกษาและแนะนำแนวทางการพัฒนา คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านในการจัดทำโครงงาน วิทยาการคอมพิวเตอร์ให้สำเร็จจุล่งและผ่านพ้นไปได้ด้วยดี

นางสาวสุชาดา มาลัยเวช รหัสนิสิต 6010450772

นางสาวฉันทิกา บำรุง รหัสนิสิต 6010451027

ผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คำนำ.....	5
กิตติกรรมประกาศ	6
สารบัญ	7
บทที่ 1 บทนำ	13
ที่มาและความสำคัญ.....	13
วัตถุประสงค์	13
อุปกรณ์ที่ใช้	13
ขั้นตอนการดำเนินงาน.....	14
ประโยชน์ที่คาดว่าจะได้รับ	14
บทที่ 2 ทฤษฎี หลักการและเทคโนโลยีที่เกี่ยวข้อง	15
บทที่ 3 วิเคราะห์และออกแบบระบบ	21
Activity Diagram	22
Use Case Diagram.....	24
Use Case Description.....	25
Entity Relationship Diagram	66
การออกแบบ User Interface.....	67
บทที่ 4 การพัฒนาระบบ	69
Templates ของหน้าเว็บ	69
การจัดส่วนของ Django Administrator หรือ Admin Dashboard.....	75
ส่วนของการจำกัดความการแสดงผลของหน้านั้นๆ.....	76

สารบัญต่อ

เรื่อง	หน้า
การกำหนดเส้นทาง	79
บทที่ 5 ผลการทดสอบระบบ	81
Front End (ฝั่งผู้ใช้)	81
Back End (ฝั่งผู้ดูแลระบบ)	89
บทที่ 6 บทสรุปและข้อเสนอแนะ.....	95
สรุปผล.....	95
ปัญหาและอุปสรรค.....	95
แนวทางการแก้ไข	95
แนวทางพัฒนา.....	96
ข้อจำกัดของระบบ	96
ข้อเสนอแนะ	96
บรรณานุกรม.....	98
ภาคผนวก	102

สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 2-1 Django Logo	15
รูปที่ 2-2 Visual Studio Code Logo	16
รูปที่ 2-3 XAMPP Logo	16
รูปที่ 2-4 phpMyAdmin logo	17
รูปที่ 2-5 Bootstrap Logo	17
รูปที่ 2-6 Stripe Logo	18
รูปที่ 2-7 Figma Logo	18
รูปที่ 2-6 DBEaver Logo	19
รูปที่ 3-1 การกำหนดขอบเขตของการซื้อสินค้าและแลกแต้มคะแนน	24
รูปที่ 3-2 ER diagram ที่สร้างโดย DBEaver	66
รูปที่ 3-3 การออกแบบส่วน navigator bar, payment, sign in และ sign up โดยใช้ Figma.....	67
รูปที่ 3-4 การออกแบบส่วน category dropdown และหมวดหมู่สินค้า โดยใช้ Figma.....	67
รูปที่ 3-5 การออกแบบส่วน cart และ order โดยใช้ Figma.....	68
รูปที่ 4-1 โค้ด HTML โครงสร้างหน้าเว็บ	70
รูปที่ 4-2 โค้ด HTML หน้า home.....	71
รูปที่ 4-3 โค้ด HTML ส่วน navigator bar	75
รูปที่ 4-4 โค้ดการทำงาน Admin Dashboard	76
รูปที่ 4-5 โค้ดการจำกัดความการทำงานหน้าแรก.....	77
รูปที่ 4-6 โค้ดจำกัดความการทำงาน หน้าตะกร้าสินค้า	78

สารบัญรูปภาพต่อ

รูปภาพ	หน้า
รูปที่ 4-7 โค้ดจำกัดความการทำงานของหน้าแบบฟอร์มลงทะเบียน.....	79
รูปที่ 4-8 โค้ดจำกัดความการลบสินค้าในหน้าตะกร้าสินค้า	79
รูปที่ 4-9 โค้ดการกำหนดเส้นทาง.....	80
รูปที่ 5-1 หน้าแรกเมื่อผู้ใช้เข้าเว็บไซต์.....	81
รูปที่ 5-2 หน้า sign in	82
รูปที่ 5-3 หน้า sign up.....	82
รูปที่ 5-4 แจ้งเตือนเมื่อผู้ใช้กรอก Username ที่ถูกใช้ลงทะเบียนไปแล้ว, กรอก email ผิดรูปแบบ, กรอก Password ไม่ตรงตามเกณฑ์การตั้งและน้อยกว่า 8 ตัวอักษร	83
รูปที่ 5-5 หน้าแรกเมื่อผู้ใช้ sign in เข้ามา	83
รูปที่ 5-6 category แสดงย่อหมวดหมู่	84
รูปที่ 5-7 ชื่อผู้ใช้แสดงย่อประวัติการสั่งซื้อ	84
รูปที่ 5-8 แจ้งเตือนของตะกร้าสินค้าเมื่อผู้ที่ยังไม่มีสินค้าในตะกร้า.....	84
รูปที่ 5-9 แสดงผลเมื่อผู้ใช้ลองค้นหาตัวอักษร “r”	85
รูปที่ 5-10 แสดงผลเมื่อผู้ใช้คลิกสินค้า Apple airpods with charging case.....	85
รูปที่ 5-11 ผู้ใช้เพิ่มสินค้าลงตะกร้า 2 รายการ	86
รูปที่ 5-12 แจ้งเตือนเมื่อผู้ใช้ต้องการลบสินค้าออกจากตะกร้า.....	86
รูปที่ 5-13 สินค้าถูกลบออกจากตะกร้าเรียบร้อยแล้ว	86
รูปที่ 5-14 ผู้ใช้กรอกรายละเอียดการจัดส่ง	87
รูปที่ 5-15 ผู้ใช้กรอกรายละเอียดบัตรเครดิตเพื่อชำระเงิน	87

สารบัญรูปภาพต่อ

รูปภาพ	หน้า
รูปที่ 5-16 แสดงหน้า Thank You! หลังทำการสั่งซื้อสำเร็จ.....	88
รูปที่ 5-17 ประวัติการสั่งซื้อสินค้า.....	88
รูปที่ 5-18 รายละเอียดใบสั่งซื้อสินค้า	88
รูปที่ 5-19 login ของ Admin.....	89
รูปที่ 5-20 หน้า home ของ Admin	90
รูปที่ 5-21 แสดงผู้ใช้ที่ทำการลงทะเบียนไว้.....	91
รูปที่ 5-22 ข้อมูลของผู้ใช้.....	91
รูปที่ 5-23 หน้าข้อมูลสินค้า	92
รูปที่ 5-24 ข้อมูลสินค้าแบบละเอียด.....	92
รูปที่ 5-25 หมวดหมู่ของสินค้าภายในร้านค้า	93
รูปที่ 5-26 คำสั่งซื้อภายในร้านค้า	93
รูปที่ 5-27 แต้มคะแนนสะสมของผู้ใช้	94
รูปที่ 5-28 Admin เช็การชำระเงินผ่านทาง Stripe API	94
รูปที่ 7-1 บัตรนัดหมายบริจาคเลือด.....	103
รูปที่ 7-2 บัตรนัดหมายบริจาคเลือด.....	103
รูปที่ 7-3 ผู้จัดทำกำลังบริจาคเลือด	103
รูปที่ 7-2 ผู้จัดทำ สุธาดา มัลลย์เวช.....	104
รูปที่ 7-2 ผู้จัดทำ ฉันทิกา บำรุง.....	104

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 3-1 Business Flow แสดงขั้นตอนการซื้อสินค้าและแลกแต้มคะแนน	23
ตารางที่ 3-2 Use Case Description: sign in ของผู้ใช้งาน	26
ตารางที่ 3-3 Use Case Description: sign in ของ Admin (Django Administrator)	31
ตารางที่ 3-4 Use Case Description: sign up ของ ผู้ใช้งาน	35
ตารางที่ 3-5 Use Case Description: ผู้ใช้เพิ่มสินค้าลงตะกร้า.....	43
ตารางที่ 3-6 Use Case Description: ผู้ใช้ลบสินค้าออกจากตะกร้า	47
ตารางที่ 3-7 Use Case Description: แต้มคะแนนสะสม.....	50
ตารางที่ 3-8 Use Case Description: ผู้ใช้ชำระเงิน	53
ตารางที่ 3-9 Use Case Description: ผู้ใช้ค้นหาในช่อง Search	59
ตารางที่ 3-10 Use Case Description: ผู้ใช้ดูประวัติสั่งซื้อ.....	62
ตารางที่ 3-11 Use Case Description: ผู้ใช้งาน sign out.....	64

บทที่ 1

บทนำ

ในบทนำจะกล่าวถึงที่มาและความสำคัญในการพัฒนาโครงการ HiBLOOD Shop Web Application นี้ซึ่งรวมถึงวัตถุประสงค์และอุปกรณ์ที่ผู้จัดทำใช้ในการพัฒนา ขั้นตอนในการดำเนินงานและประโยชน์ของผู้ใช้ web application ที่จะได้รับ

1.1. ที่มาและความสำคัญ

ผู้บริจาคเลือดส่วนใหญ่ ต้องการบริจาคเลือดเนื่องจากมีความเอื้อเฟื้อและต้องการช่วยเหลือเพื่อนมนุษย์ แต่เนื่องด้วยสถานการณ์โรคระบาดจากไวรัสโคโรนา(Covid-19)ทำให้จำนวนผู้บริจาคเลือดลดลงเป็นอย่างมากถึงขั้นวิกฤต ส่งผลให้เลือดในศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทยนั้น มีไม่เพียงพอต่อการใช้เพื่อรักษาให้กับผู้ป่วยที่ต้องการเลือด เกิดการขาดแคลนบ่อยครั้ง ผู้จัดทำจึงคิดใช้จำนวนครั้งของการไปบริจาคเลือดไปเป็นคะแนนสะสมและนำคะแนนมาเป็นส่วนลดราคาสິงของภายในเว็บแอปพลิเคชันร้านค้าของเรา การสร้างเว็บแอปพลิเคชันนี้เปรียบเสมือนแทนการขอบคุณและให้กำลังใจกับผู้บริจาคเลือด ที่มาบริจาคเลือดอย่างต่อเนื่องหลายๆ ครั้ง และเป็นการกระตุ้นทางอ้อมให้เกิดเป้าหมายนอกเหนือจากการเอื้อเฟื้อหรือต้องการช่วยเหลือเพื่อนมนุษย์ รวมถึงให้ผู้ที่เคยบริจาคเลือดแล้วหยุดบริจาคได้กลับมาบริจาคต่อหรือผู้ไม่เคยบริจาคเลือดเลยอยากลองบริจาคเลือดในครั้งแรกและทำการบริจาคอย่างต่อเนื่องจนครบเป้าหมายของแต้มคะแนนที่ต้องการนั้น

ผู้จัดทำได้กำหนดให้ การบริจาค 1 ครั้ง เท่ากับ 450 แต้มคะแนนและ 1 แต้มคะแนน เท่ากับ 1 บาท โดยแต้มคะแนนของเราไม่มีวันหมดอายุหรือแต้มคะแนนสะสมไม่ถึงก็สามารถจ่ายส่วนต่างที่ไม่ได้ลดราคาสามารถดำเนินการชำระเงินผ่านบัตรเครดิต/เครดิต ก็ทำได้อย่างปลอดภัย

1.2. วัตถุประสงค์

- เพื่อให้ web application เป็นสื่อกลางขอบคุณผู้บริจาคที่บริจาคเลือดอย่างต่อเนื่อง
- เพื่อพัฒนา web application มาใช้ในการกระตุ้นทางอ้อมให้ผู้บริจาคอยากมาบริจาคเลือดอย่างต่อเนื่อง

1.3. อุปกรณ์ที่ใช้

- โปรแกรม Django framework ใช้ในการสร้าง web application
- โปรแกรม Visual studio code ในการเขียนโปรแกรมภาษา python
- โปรแกรม xampp เพื่อใช้ phpMyAdmin ในการเก็บฐานข้อมูล

- เครื่องมือเพิ่มเติม Bootstrap ในการตกแต่งหน้า UI, Stripe API ในการชำระเงิน
- โปรแกรม Figma ในการ design web application
- โปรแกรม DBeaver ในการสร้าง Er diagram

1.4. ขั้นตอนการดำเนินงาน

- เลือกหัวข้องานวิจัย
- ศึกษาค้นคว้าหาข้อมูลจากแหล่งข้อมูลต่างๆ และขอคำแนะนำจากอาจารย์ที่ปรึกษางานวิจัย
- วางแผนการทำงาน
- ออกแบบและจัดทำ web application
- ทำการทดสอบ web application
- จัดทำรายงาน
- นำเสนองาน

1.5. ประโยชน์ที่คาดว่าจะได้รับ

- เพื่อเกิดการกระตุ้นทางอารมณ์ให้มีเป้าหมายสามารถทำให้ผู้บริจาคอยากบริจาคเลือดอย่างต่อเนื่อง
- สามารถเป็นสื่อกลางในการขอบคุณให้กับกับผู้บริจาคเลือดได้
- เกิดการบริจาคเลือดเพิ่มมากขึ้นจากผู้ที่ไม่เคยบริจาคลองบริจาคเลือดและผู้ที่ยุติบริจาคไปกลับมาบริจาคอีกครั้ง

บทที่ 2

ทฤษฎี หลักการ และเทคโนโลยีที่เกี่ยวข้อง

ในการจัดทำโครงงาน เว็บแอปพลิเคชันไฮบลัด (HiBLOOD Shop Web Application) ใช้ Figma ในการออกแบบ User Interface พัฒนาโดยใช้ภาษา Python และใช้ Django framework ช่วยในการเขียนโปรแกรมสร้าง web application โดยเขียนผ่านทางเครื่องมือ Visual Studio Code เก็บฐานข้อมูลหรือ Database ใน phpMyAdmin ใช้ Bootstrap ตกแต่งหน้า Web Application เพื่อความสวยงามและให้ผู้ใช้สามารถใช้งานง่าย มีการใช้ Stripe API ช่วยในเรื่องชำระเงินเพื่อผ่านบัตรเครดิต/เครดิต ซึ่งรายละเอียดของโปรแกรมต่างๆจะกล่าวต่อไป

2.1. Django (จิงโก้)

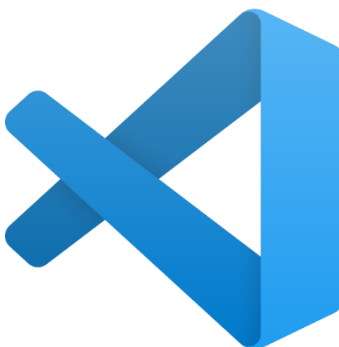


รูปที่ 2-1 Django Logo

ที่มา: <https://www.djangoproject.com/community/logos/>

Django (จิงโก้ ไม่ออกเสียงตัว D) เป็น framework ที่ใช้ในการสร้าง Web Application ในทางฝั่งของ Back-end พัฒนาด้วยภาษา Python ใน framework จะมีส่วนประกอบที่จำเป็นพื้นฐาน เช่น การเชื่อมต่อฐานข้อมูล, render ข้อมูล ในทางฝั่งของ Front-end⁽⁴⁾ มีระบบจัดการ Database ที่ออกแบบมาให้ใช้งานง่าย มี ระบบ Admin สำเร็จรูปให้ใช้⁽⁵⁾ และยังสามารถปรับแต่งให้ตรงความต้องการได้ว่าจะใช้หรือไม่ใช้ส่วนใด

2.2. Visual studio code



รูปที่ 2-2 Visual Studio Code Logo

ที่มา:https://th.wikipedia.org/wiki/%E0%B9%84%E0%B8%9F%E0%B8%A5%E0%B9%8C:Visual_Studio_Code_1.35_icon.svg

VS Code หรือ Visual Studio Code จากบริษัทไมโครซอฟต์ เป็นโปรแกรมประเภท Editor ใช้ในการแก้ไขโค้ดที่มีขนาดเล็กแต่มีประสิทธิภาพสูง เป็น Open Source โปรแกรมจึงสามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานหลาย platform รองรับการใช้งานทั้งบน Windows, macOS และ Linux รองรับหลายภาษาทั้ง JavaScript, Typescript และ Node.js ในตัวและสามารถเชื่อมต่อกับ Git ได้ง่าย สามารถนำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือและส่วนขยายต่าง ๆ ให้เลือกใช้มากมาย รองรับการใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP⁽⁶⁾

2.3. Xampp



รูปที่ 2-3 XAMPP Logo

ที่มา:<http://www.ruyoung.com/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%95%E0%B8%B4%E0%B8%94%E0%B8%95%E0%B8%B1%E0%B9%89%E0%B8%87-xampp/>

XAMPP เป็นคำย่อของโปรแกรมที่ประกอบด้วย Apache, MySQL, PHP และ Perl เป็นเว็บเซิร์ฟเวอร์จำลองที่พัฒนาขึ้นเพื่อใช้ในการทดสอบหรือทดลองเว็บไซต์ XAMPP เป็นโปรแกรมประเภทที่เรียกว่า cross-platform คือใช้ได้ทั้งบนระบบปฏิบัติการ Windows, Mac OS และ Linux⁽⁷⁾

2.4. phpMyAdmin



รูปที่ 2-4 phpMyAdmin Logo

ที่มา: https://www.bestinternet.co.th/single_blog.php?id=10&phpMyAdmin%20%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3

phpMyAdmin คือโปรแกรม PHP ที่ถูกออกแบบเพื่อใช้บริหารจัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ซึ่งหากเราใช้โปรแกรมนี้อาจจะทำให้เราสามารถจัดการฐานข้อมูลได้ง่ายขึ้นเราสามารถที่จะใช้โปรแกรมนี้อาจที่จะทำการสร้างฐานข้อมูลใหม่ (New Database) สร้างตารางข้อมูล (Table) ทำการประมวลผล (Query) ข้อมูลที่เราต้องการและมีฟังก์ชันที่ใช้บริหารจัดการอื่นๆ เช่น เพิ่ม (Insert), ลบ (Delete), อัปเดต (Update), ค้นหา (Browse), นำเข้าหรือนำออกฐานข้อมูล (Import and Export Database), ลบทิ้ง (Drop) และอื่นๆ อีกมากมายให้ใช้งาน⁽⁸⁾

2.5. Bootstrap

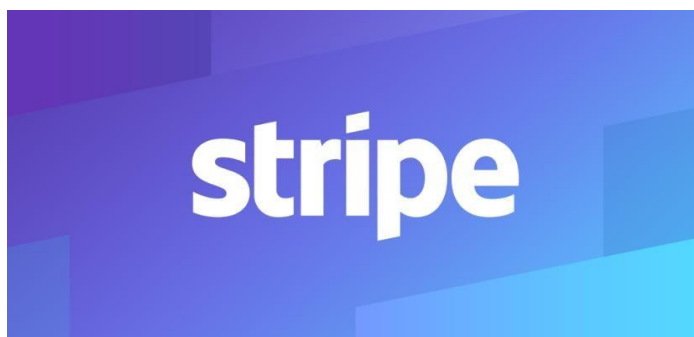


รูปที่ 2-5 Bootstrap Logo

ที่มา: <https://medium.com/@darwich.ziad/create-bootstrap-logo-in-css-979797d343aa>

Bootstrap คือ Front-end Framework ยี่ห้อนึงที่ช่วยให้เราสามารถสร้างหน้าเว็บให้ตรงตามแบบที่เราต้องการได้ง่ายขึ้น เพราะ Bootstrap มีทั้งระบบ grid ที่ช่วยเรื่องการวาง layout ที่รองรับในแบบ responsive และมี component สำเร็จรูปให้ใช้ ถ้าเราอยากได้ของที่ต้องใช้บ่อยๆ ⁽⁹⁾

2.6. Stripe API



รูปที่ 2-6 Stripe Logo

ที่มา: <https://medium.com/wineofbits/stripe-101-how-stripe-payment-api-works-59f3149ebe3f>

Stripe คือระบบตัวกลางที่ไว้ใช้รับเงินจากลูกค้าผ่านบัตรเครดิต/เครดิตเช่นเดียวกับPaypal เพียงแต่ว่ามาในรูปแบบที่ดูง่ายกว่าเพียงแค่คัดลอกโค้ดซึ่งเชื่อมต่อกับระบบของ Stripe ผ่าน API ไปวางตามตำแหน่งต่างๆ ของหน้าเว็บไซต์หรือหน้าแอปที่ต้องการเก็บเงินก่อนเข้ารับบริการหรือเพื่อซื้อสินค้า Stripe ทำงานได้กับภาษาโปรแกรม เช่น cURL, Ruby, Python, PHP, Java, Node และ Go ซึ่งเหล่านักพัฒนาสามารถหยิบเครื่องมือเหล่านี้ไปใส่เข้ากับแอปพลิเคชันหรือเว็บไซต์ของตนได้ทันทีจากนั้นจึงเชื่อมกับ Account Stripe ของผู้ใช้ซึ่งเป็นประตูกระเป๋าสตางค์นับยอดเงินเข้าออก ⁽¹⁰⁾

2.7. Figma



รูปที่ 2-7 Figma Logo

ที่มา: <https://medium.com/teamappman/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-figma-%E0%B8%AD%E0%B8%AD%E0%B8%81%E0%B9%81%E0%B8%9A%E0%B8%9A-ui-basic-6675da0a0926>

Figma เป็น Design Tool Website และ Application ที่สามารถเข้าไปออกแบบงานประเภท User Interface ได้ทันทีโดยไม่ต้องทำการลงโปรแกรมใดๆ การบันทึกข้อมูลก็ไม่กินพื้นที่บนเครื่องของผู้ใช้งานด้วย อาจจะมาแทนโปรแกรมอย่าง Sketch หรือ Adobe XD ในการออกแบบ User Interface กันได้เลย และยังให้เข้าไปใช้งานได้ฟรี สามารถเชิญผู้อื่นเข้ามาดูหรือร่วมแก้ไขไฟล์เดียวกันได้อีกด้วย⁽¹¹⁾

2.8. DBeaver



รูปที่ 2-6 DBeaver Logo

ที่มา: <https://ponggun.medium.com/%E0%B8%AA%E0%B8%A3%E0%B9%89%E0%B8%B2%E0%B8%87-entity-relationship-diagrams-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-ssms-dbeaver-draw-io-dbdiagram-io-765e77272605>

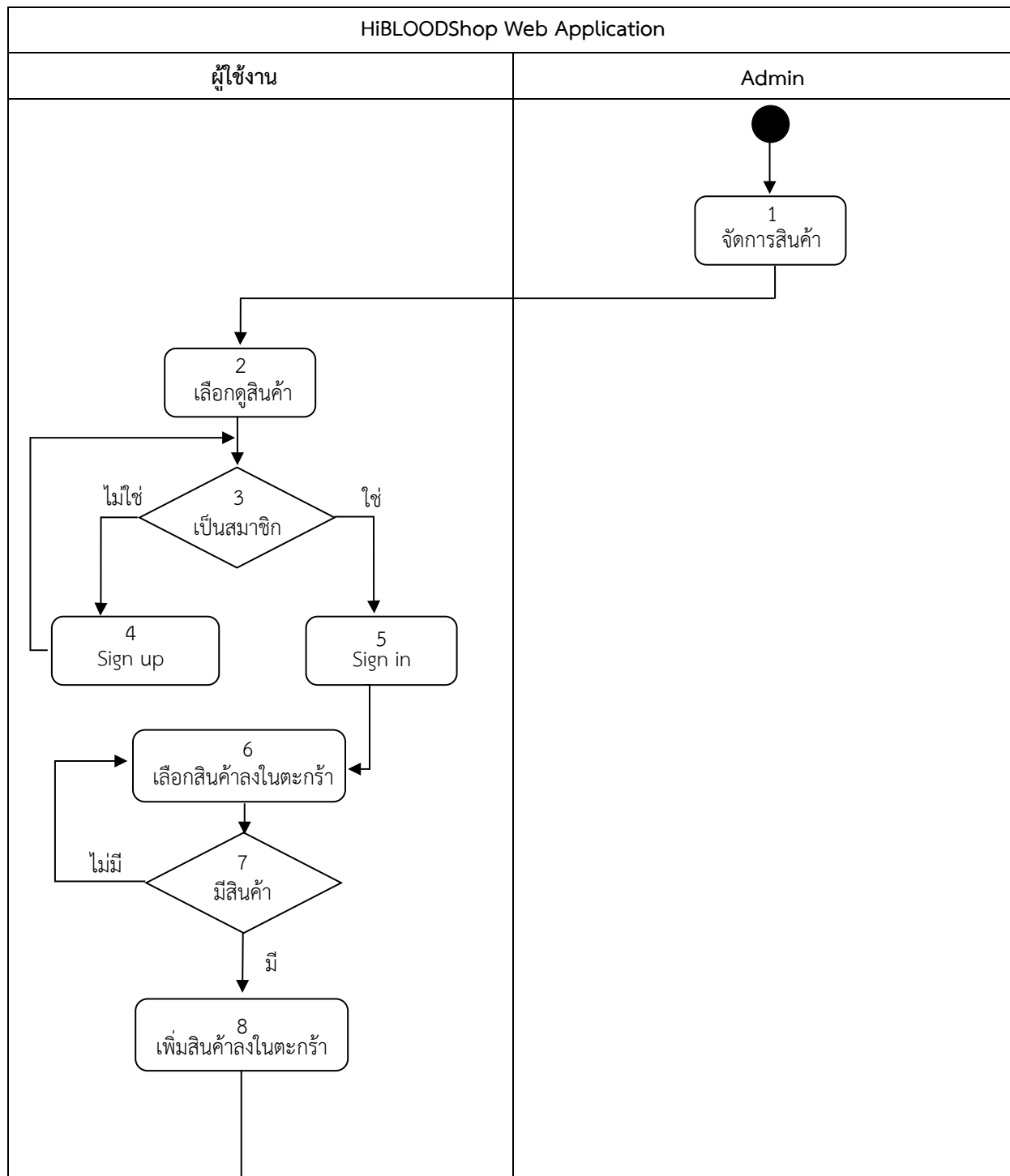
DBeaver เป็นซอฟต์แวร์ที่ทำหน้าที่เป็นเครื่องมือฐานข้อมูลสากล มีไว้สำหรับนักพัฒนาฐานข้อมูลและ
ผู้ดูแลระบบ DBeaver มีอินเทอร์เฟซผู้ใช้ที่ออกแบบมาอย่างดี แพลตฟอร์มนี้ใช้กรอบโอเพนซอร์สและอนุญาตให้
เขียนส่วนขยายได้หลายแบบ รวมถึงเข้ากันได้กับฐานข้อมูลใดๆด้วย รวมถึงการสนับสนุนไคลเอนต์ MySQL และ
Oracle ตั้งแต่ การจัดการไดรเวอร์ตัวแก้ไข SQLและการจัดรูปแบบ DBeaver เป็นแอปพลิเคชันข้ามแพลตฟอร์ม
เนื่องจากรองรับแพลตฟอร์ม MacOS, Windows และ Linux⁽¹²⁾

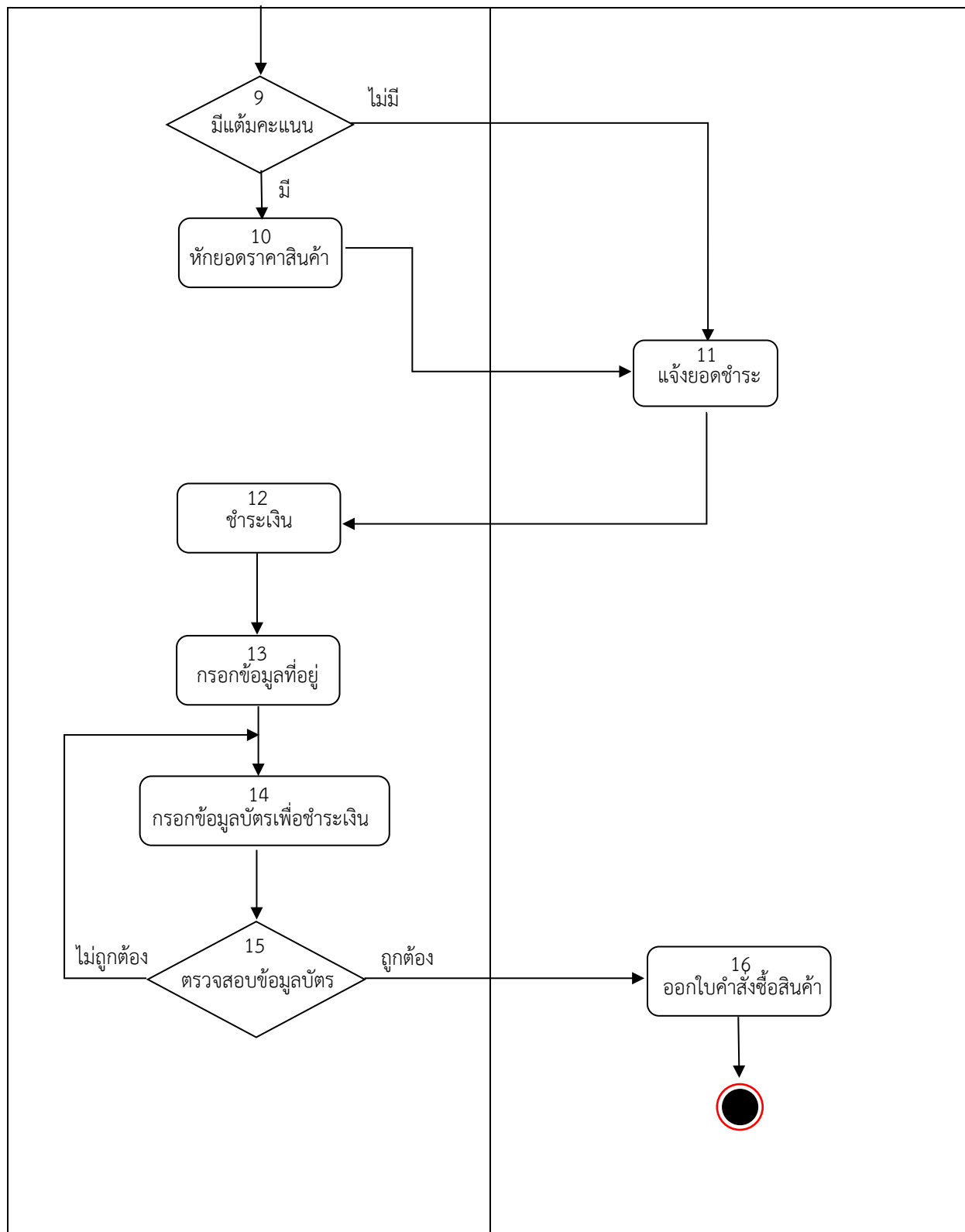
บทที่ 3

วิเคราะห์และออกแบบระบบ

ในบทนี้จะกล่าวถึง Flowchart Diagram ที่ใช้อธิบายการทำงานของระบบ ว่าระบบนี้ทำงานอย่างไร เริ่มจากไหนและจบที่ไหน เป็นแบบ Step by step เพราะช่วยลำดับขั้นตอนของการทำงานของระบบได้อย่างถูกต้อง, สร้างความเข้าใจที่ตรงกันของทีม และผู้ที่เกี่ยวข้อง, ช่วยให้เราสามารถวิเคราะห์การทำงานของระบบได้ดีขึ้น ถัดไปคือ Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นว่าระบบมีการทำงานหรือมีหน้าที่ใดบ้าง โดยมีสัญลักษณ์รูปวงรีแทน Use Case และสัญลักษณ์รูปคน (Stick Man Icon) แทน Actor และ Use Case Description เป็นคำอธิบายระบบแบบละเอียด และส่วนสุดท้ายของบทนี้เป็นการออกแบบ User Interface ของระบบเพื่อให้เห็นภาพรวมที่ใช้ในการพัฒนา

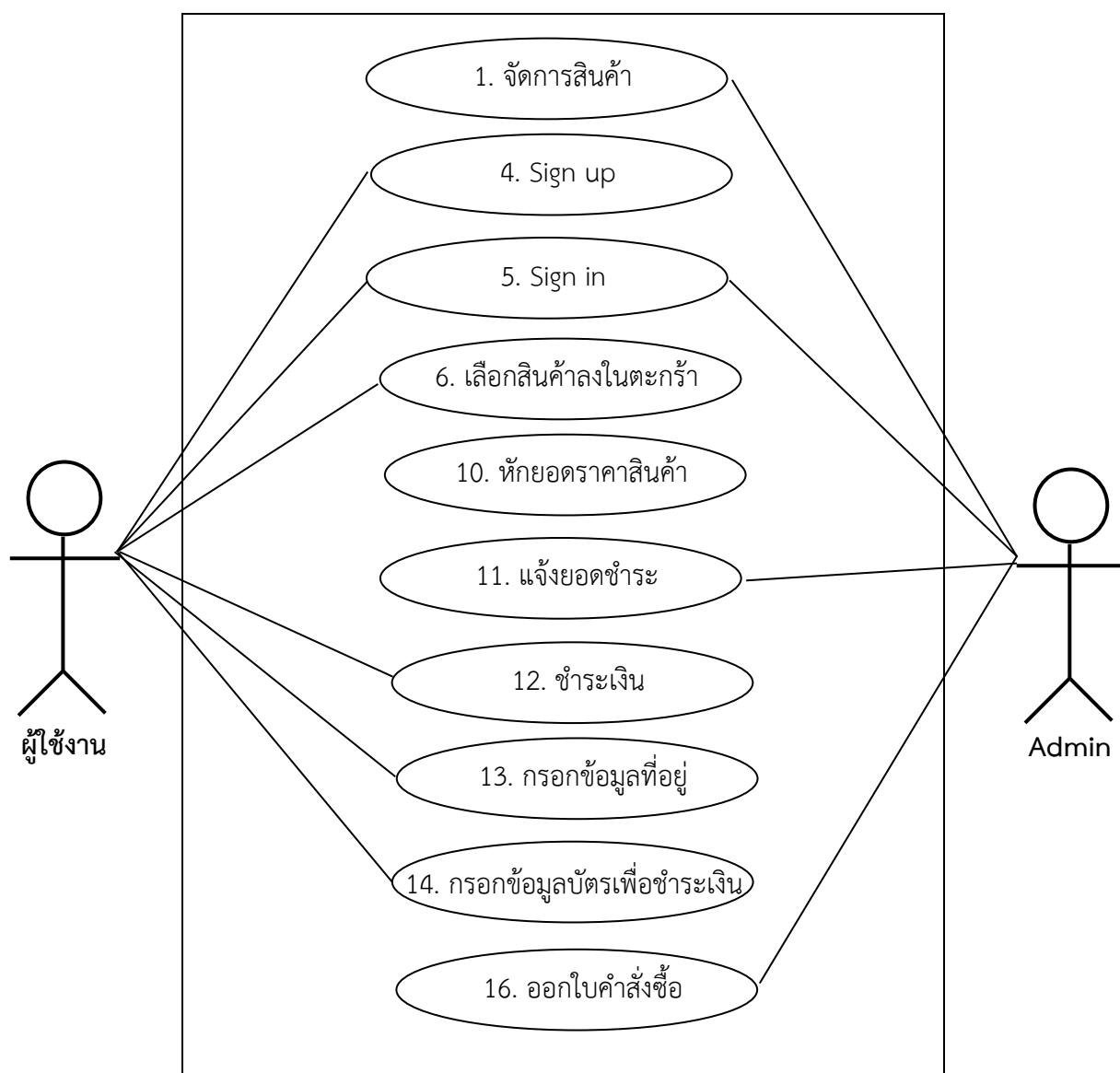
3.1. Activity Diagram แสดงขั้นตอนการซื้อสินค้าและแลกแต้มคะแนน





ตารางที่ 3-1 Business Flow แสดงขั้นตอนการซื้อสินค้าและแลกแต้มคะแนน

3.2. Use case diagram



รูปที่ 3-1 การกำหนดขอบเขตของการซื้อสินค้าและแลกแต้มคะแนน

3.3. Use case Description

➤ Use Case Description: sign in ของผู้ใช้งาน

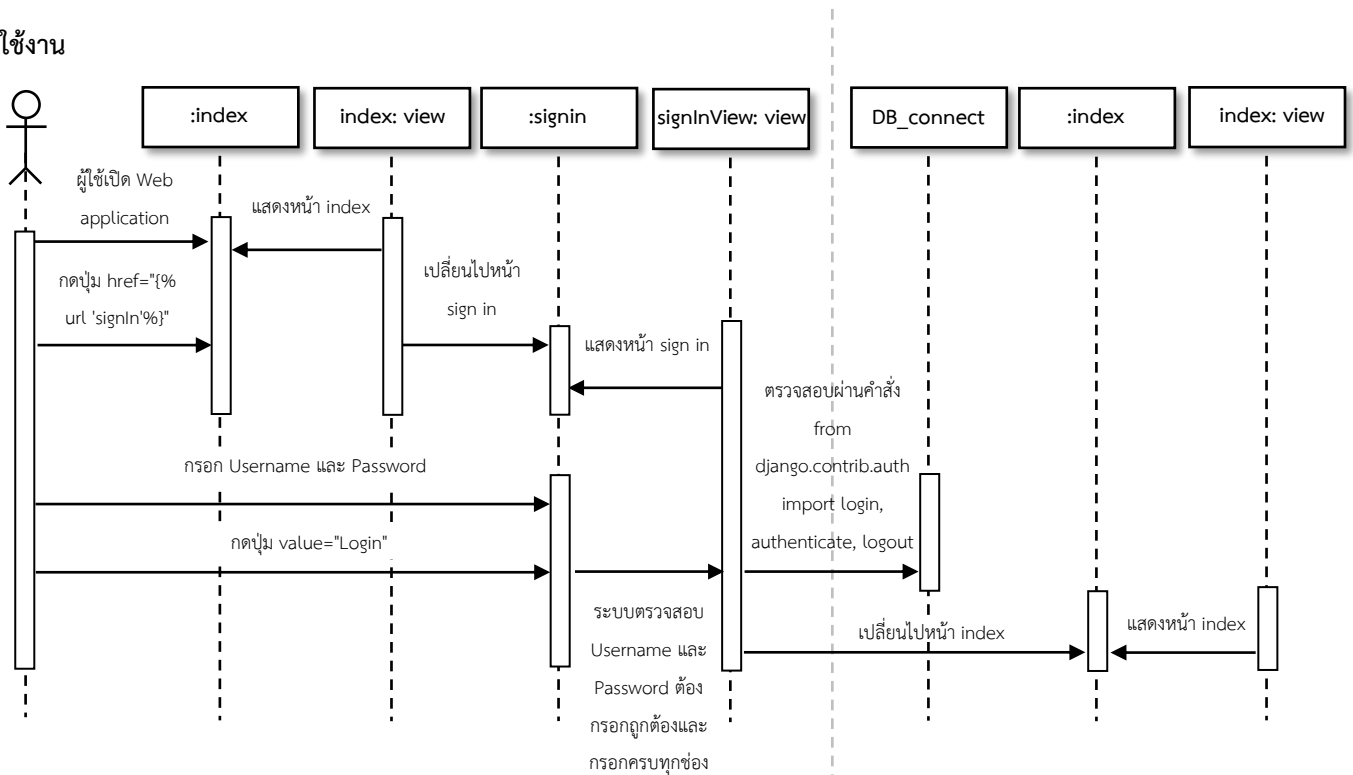
Use case	1	
Use case name	sign in	
Actor	ผู้ใช้งาน	
Description	เข้าสู่ระบบเพื่อเลือกซื้อสินค้า	
Precondition	ต้องการซื้อสินค้าและทำการ sign up แล้ว	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่ม sign in 3. กรอก Username และ Password 4. กดปุ่ม Log in	2. แสดงหน้า sign in 5. ระบบตรวจสอบ Username และ Password ต้องกรอกถูกต้องและกรอกครบทุกช่อง 6. ตรวจสอบผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout
Post condition	เปลี่ยนไปหน้า Home ของผู้ใช้งาน	
Alternative Course 1	Actor Action	System Response
	A3.1. ไม่กรอก Username และ Password หรือช่องใดช่องหนึ่งว่าง A3.2. กดปุ่ม Log in	A3.3. ระบบตรวจสอบ Username

		<p>และ Password ต้องกรอกถูกต้อง และกรอกครบทุกช่อง</p> <p>A3.4. ตรวจสอบ ผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout</p> <p>A3.5. ไม่สามารถตรวจสอบได้</p> <p>A3.6. ระบบแสดงข้อความ alert “กรุณากรอกข้อมูลช่องนี้”</p>
Alternative Course 2	Actor Action	System Response
	<p>B3.1. กรอก Username และ Password ผิด หรือ กรอก ช่องใดช่องหนึ่งผิด</p> <p>B3.2. กดปุ่ม Log in</p>	<p>B3.3. ระบบตรวจสอบ Username และ Password ต้องกรอกถูกต้อง และกรอกครบทุกช่อง</p> <p>B3.4. ตรวจสอบ ผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout</p> <p>B3.5. ไม่สามารถตรวจสอบได้</p> <p>B3.6. ระบบแสดงข้อความ alert “Please enter the a correct username and password. Not that both fields may be case-sensitive”</p>

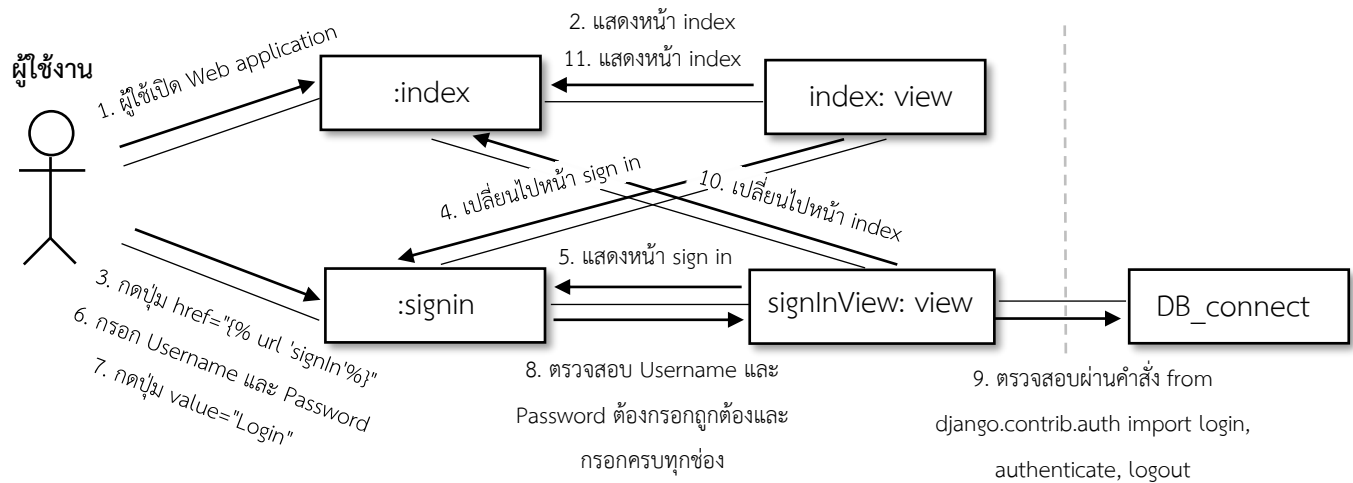
ตารางที่ 3-2 Use Case Description: sign in ของผู้ใช้งาน

➤ Sequence diagram: sign in ของผู้ใช้งาน

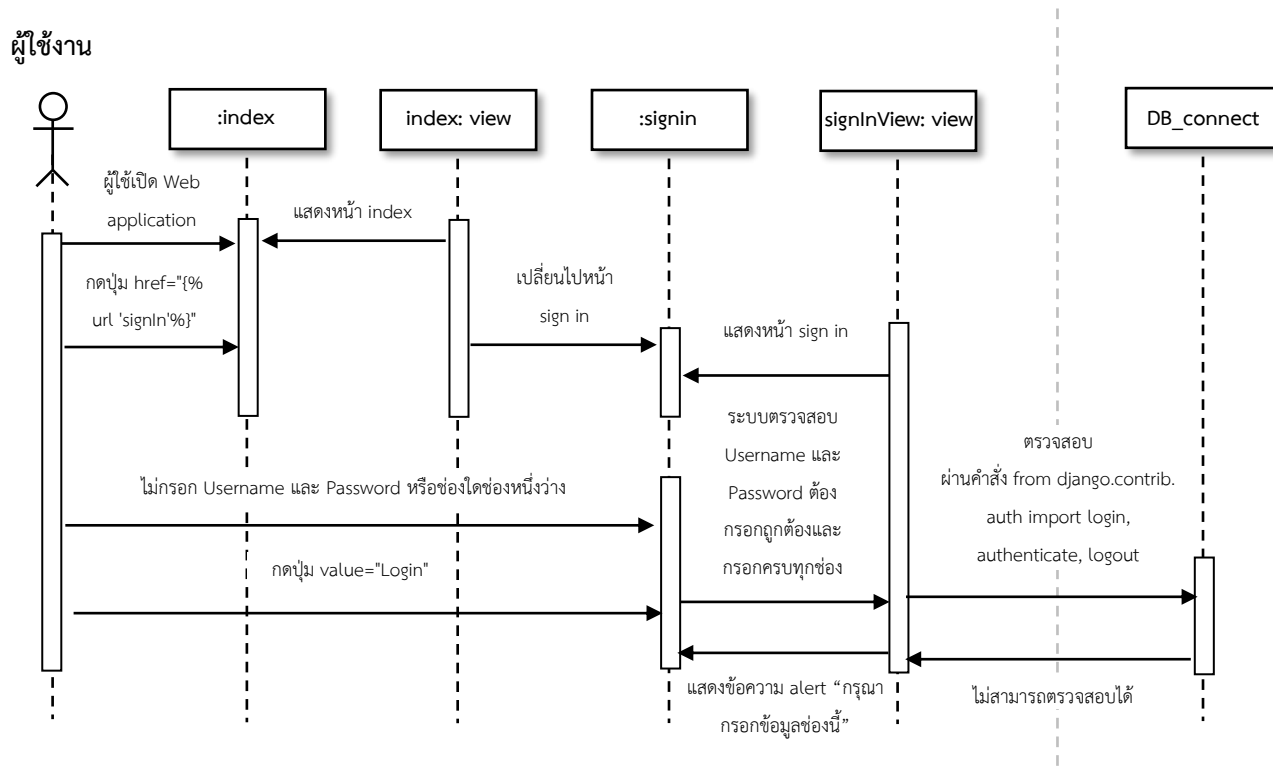
ผู้ใช้งาน



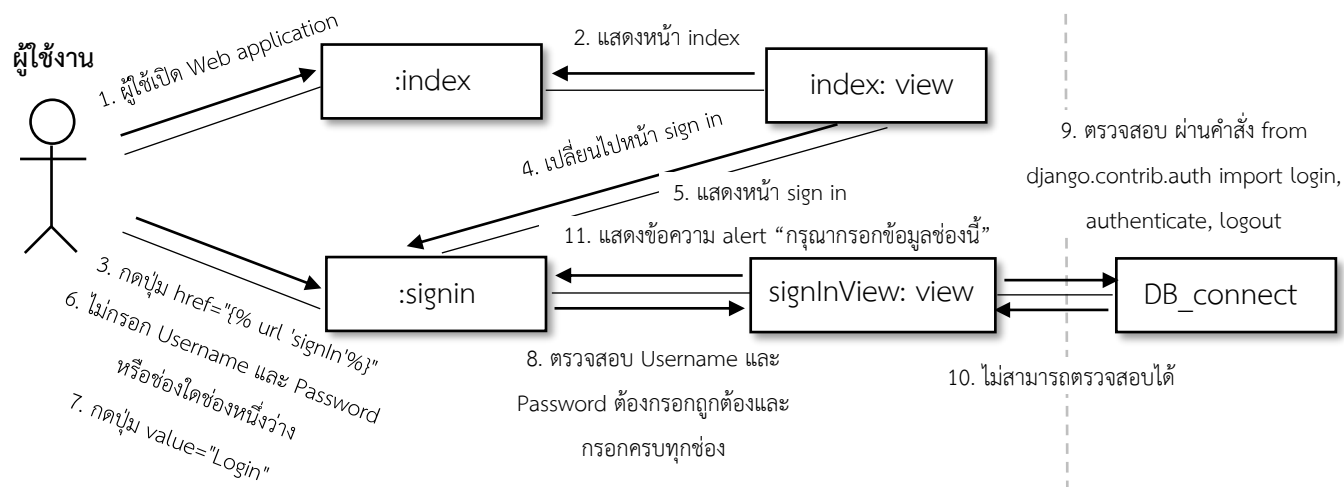
➤ Collaboration Diagram: sign in ของผู้ใช้งาน



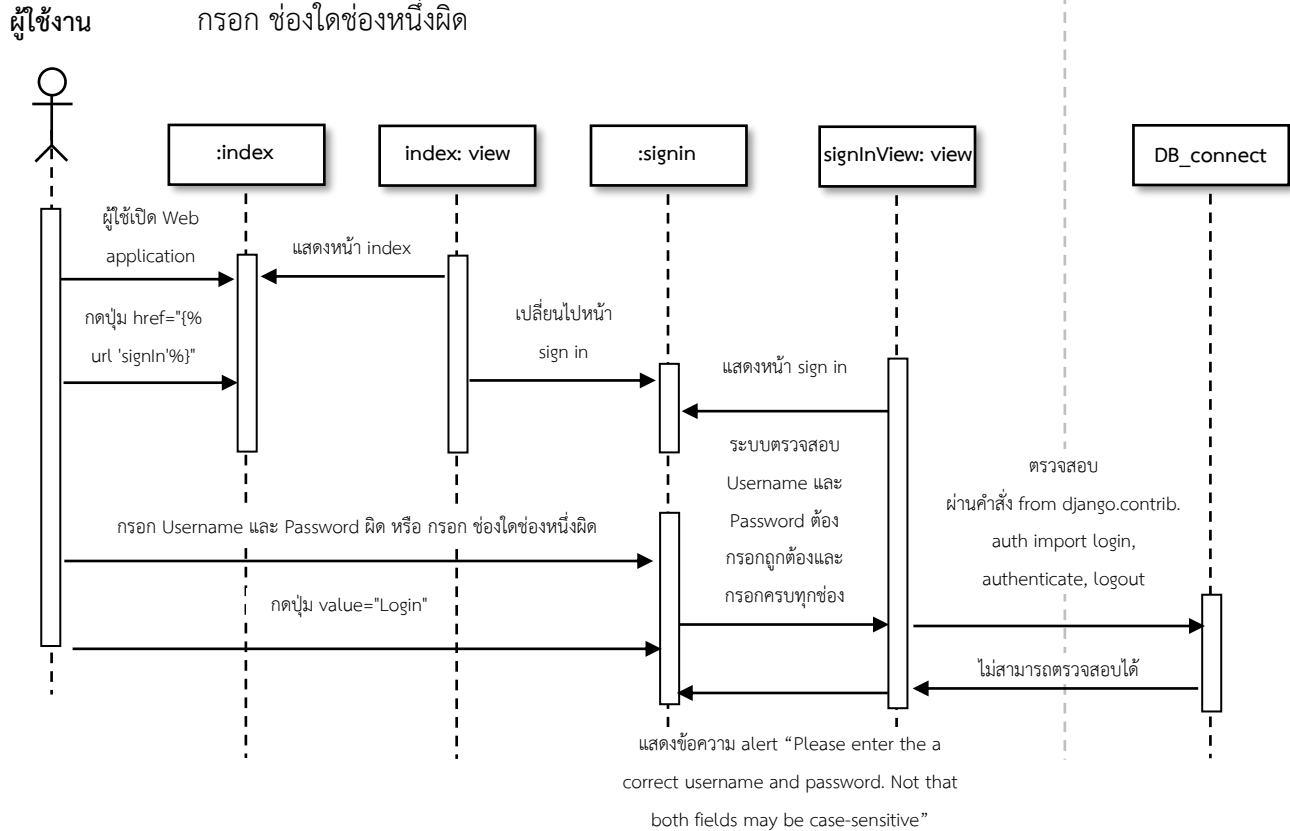
- **Sequence diagram:** sign in ของผู้ใช้งาน กรณี ไม่กรอก Username และ Password หรือช่องใดช่องหนึ่งว่าง



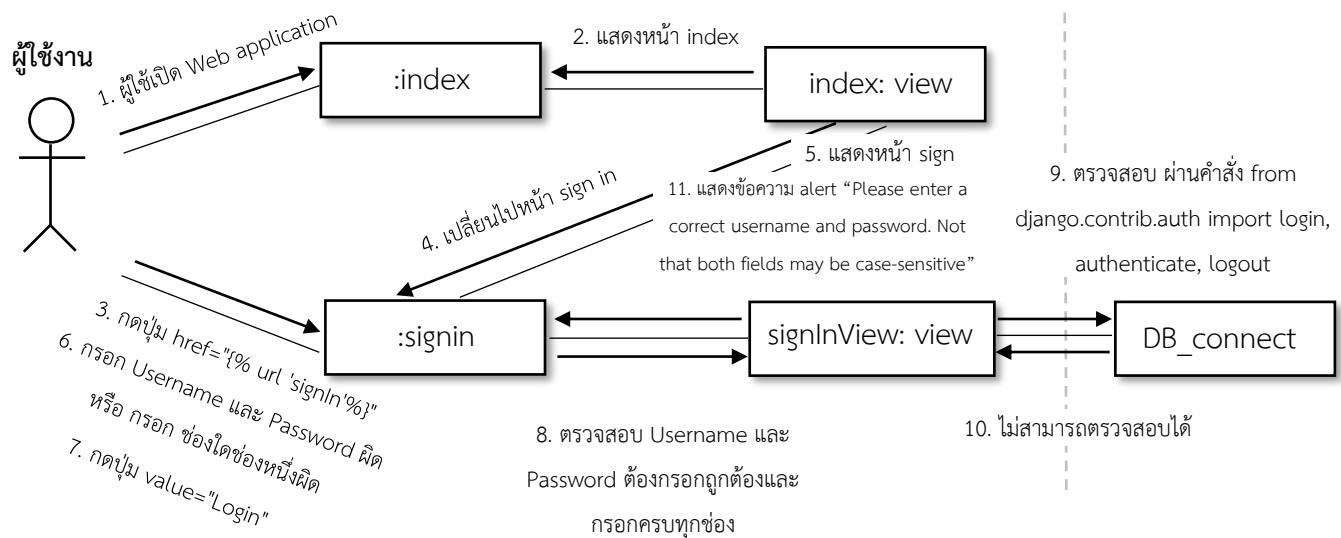
- **Collaboration Diagram:** sign in ของผู้ใช้งาน กรณี ไม่กรอก Username และ Password หรือช่องใดช่องหนึ่งว่าง



- **Sequence diagram:** sign in ของผู้ใช้งาน กรณี กรอก Username และ Password ผิด หรือ กรอก ช่องใดช่องหนึ่งผิด



- **Collaboration Diagram:** sign in ของผู้ใช้งาน กรณี กรอก Username และ Password ผิด หรือ กรอก ช่องใดช่องหนึ่งผิด



➤ Use Case Description: sign in ของ Admin (Django Administrator)

Use case	2	
Use case name	sign in	
Actor	Admin	
Description	เข้าสู่ระบบเพื่อจัดการสินค้า ดูข้อมูลลูกค้า และรับ Order	
Precondition	มี super user แล้ว	
Normal Flow	Actor Action	System Response
	1. Admin เข้าหน้า Django administration ผ่าน path../admin 3. กรอก Username และ Password 4. กดปุ่ม Log in	2. Django แสดงหน้า Admin 5. ระบบ Django ตรวจสอบ Username และ Password ต้องกรอกถูกต้องและกรอกครบทุกช่อง 6. ตรวจสอบผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout
Post condition	เปลี่ยนไปหน้า Home ของ Admin	
Alternative Course 1	Actor Action	System Response
	A3.1. ไม่กรอก Username และ Password หรือช่องใดช่องหนึ่งว่าง	A3.3. ระบบ django ตรวจสอบ Username และ Password

	A3.2. กดปุ่ม Log in	<p>ต้องการกรอกถูกต้องและกรอกครบทุกช่อง</p> <p>A3.4. ตรวจสอบ ผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout</p> <p>A3.5. ไม่สามารถตรวจสอบได้</p> <p>A3.6. ระบบแสดงข้อความ alert “กรุณากรอกข้อมูลช่องนี้”</p>
Alternative Course 2	Actor Action	System Response
	<p>B3.1. กรอก Username และ Password ผิด หรือ กรอก ช่องใดช่องหนึ่งผิด</p> <p>B3.2. กดปุ่ม Log in</p>	<p>B3.3. ระบบ Django ตรวจสอบ Username และ Password ต้องการกรอกถูกต้องและกรอกครบทุกช่อง</p> <p>B3.4. ตรวจสอบ ผ่านคำสั่ง from django.contrib.auth import login, authenticate, logout</p> <p>B3.5. ไม่สามารถตรวจสอบได้</p> <p>B3.6. ระบบแสดงข้อความ alert “Please enter the correct username and password for a staff account. Not that both fields may be case-sensitive”</p>

ตารางที่ 3-3 Use Case Description: sign in ของ Admin (Django Administrator)

➤ Use Case Description: sign up ของ ผู้ใช้งาน

Use case	3	
Use case name	Sign up	
Actor	ผู้ใช้งาน	
Description	สมัครสมาชิกโดยการกรอกแบบฟอร์มลงทะเบียน	
Precondition	ไม่เคย sign up/ ยังไม่มี account	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่ม sign up 3. กรอกข้อมูล {Frist name, Last name, Username, Email, Password, Password confirmation} 4. กดปุ่ม Register	2. ระบบแสดงหน้าแบบฟอร์มลงทะเบียน 5. ระบบ Django ตรวจสอบต้องกรอก {Firstname, Lastname, Username, Email, Password, Password confirmation} ถูกต้องและครบ 6. สร้าง User accout ผ่านคำสั่ง fromdjango.contrib.auth.forms import UserCreationForm
Post condition	ผู้ใช้สามารถลงทะเบียนได้และเปลี่ยนไปหน้า sign in	
Alternative Course 1	Actor Action	System Response

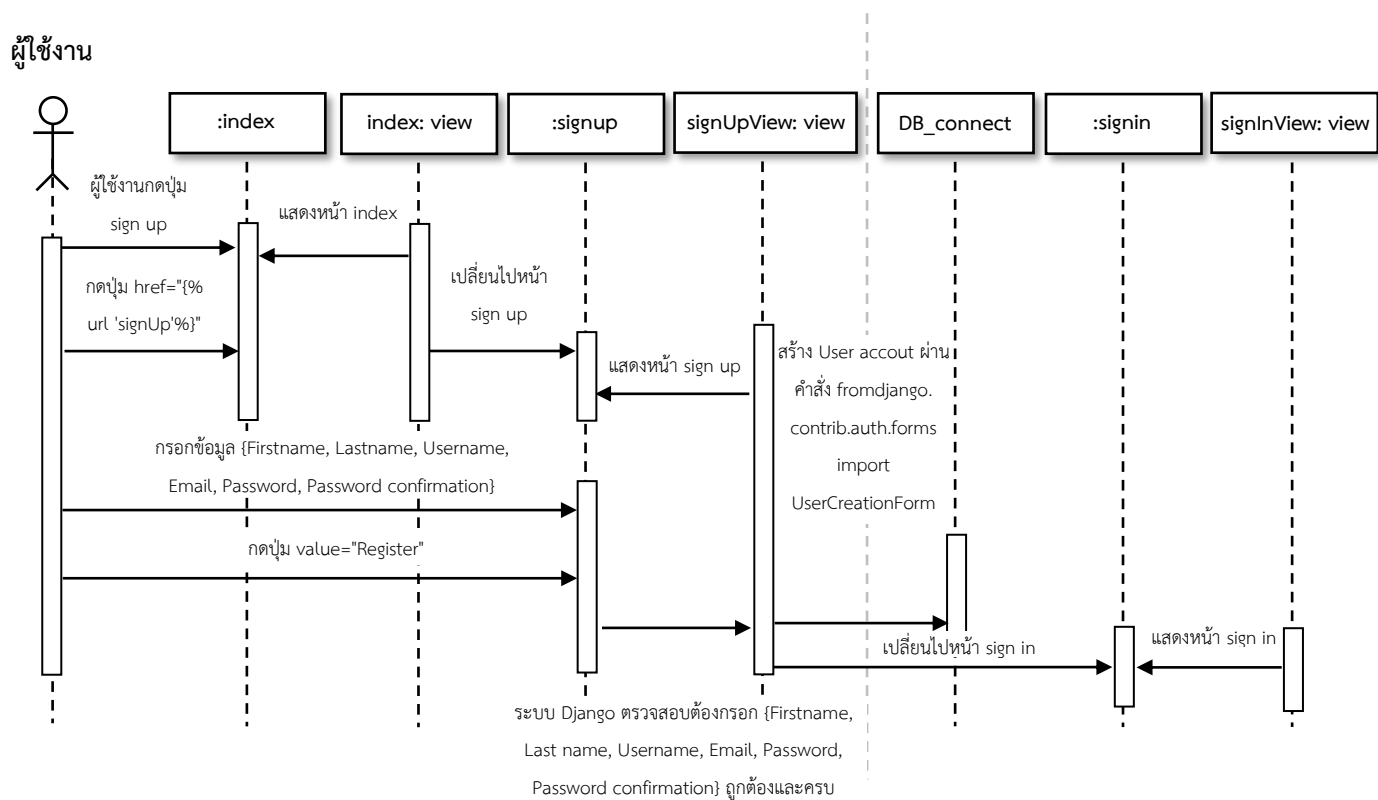
	<p>A3.1. ไม่กรอกข้อมูลทุกช่อง หรือช่องใดช่องหนึ่งว่าง</p> <p>A3.2. กดปุ่ม Register</p>	<p>A3.3. ระบบ Django ตรวจสอบ ต้องกรอก {Firstname, Last name, Username, Email, Password, Password confirmation} ถูกต้องและครบ</p> <p>A3.4. ตรวจสอบผ่านคำสั่ง from django.contrib.auth.forms import UserCreationForm</p> <p>A3.5. ไม่สามารถลงทะเบียนได้</p> <p>A3.6. ระบบแสดงข้อความ alert “กรุณากรอกข้อมูลช่องนี้”</p>
Alternative Course 2	Actor Action	System Response
	<p>B3.1. กรอก Username ที่ไม่ใช่ข้อความ, ตัวเลข, “@”, “.”, “+”, “-”, “_”</p> <p>B3.2. กดปุ่ม Register</p>	<p>B3.3. ระบบ Django ตรวจสอบ ต้องกรอก {Firstname, Last name, Username, Email, Password, Password confirmation} ถูกต้องและครบ</p> <p>B3.4. ตรวจสอบผ่านคำสั่ง from django.contrib.auth.forms import UserCreationForm</p> <p>B3.5. ไม่สามารถลงทะเบียนได้</p> <p>B3.6. ระบบแสดงข้อความ alert “Enter a valid username. This value may contain only letter, number, and @/./+/_ character.”</p>
Alternative Course 3	Actor Action	System Response

	<p>C3.1. กรอก email ไม่ใส่ “@”</p> <p>C3.2. กดปุ่ม Register</p>	<p>C3.3. ระบบ Django ตรวจสอบ ต้องกรอก {Firstname, Last name, Username, Email, Password, Password confirmation} ถูกต้องและครบ</p> <p>C3.4. ตรวจสอบผ่านคำสั่ง from django.contrib.auth.forms import UserCreationForm</p> <p>C3.5. ไม่สามารถลงทะเบียนได้</p> <p>C3.6. ระบบแสดงข้อความ alert “กรุณารวม ‘@’ ในที่อยู่อีเมล (ชื่อUsername)’กำลังขาด ‘@’ ”</p>
Alternative Course 4	Actor Action	System Response
	<p>D3.1. กรอก Email ที่ไม่ใช่ “@gmail.com”</p> <p>D3.2. กดปุ่ม Register</p>	<p>D3.3. ระบบ Django ตรวจสอบ ต้องกรอก {Firstname, Last name, Username, Email, Password, Password confirmation} ถูกต้องและครบ</p> <p>D3.4. ตรวจสอบผ่านคำสั่ง from django.contrib.auth.forms import UserCreationForm</p> <p>D3.5. ไม่สามารถลงทะเบียนได้</p> <p>D3.6. ระบบแสดงข้อความ alert “Enter a valid email address.”</p>
Alternative Course 5	Actor Action	System Response

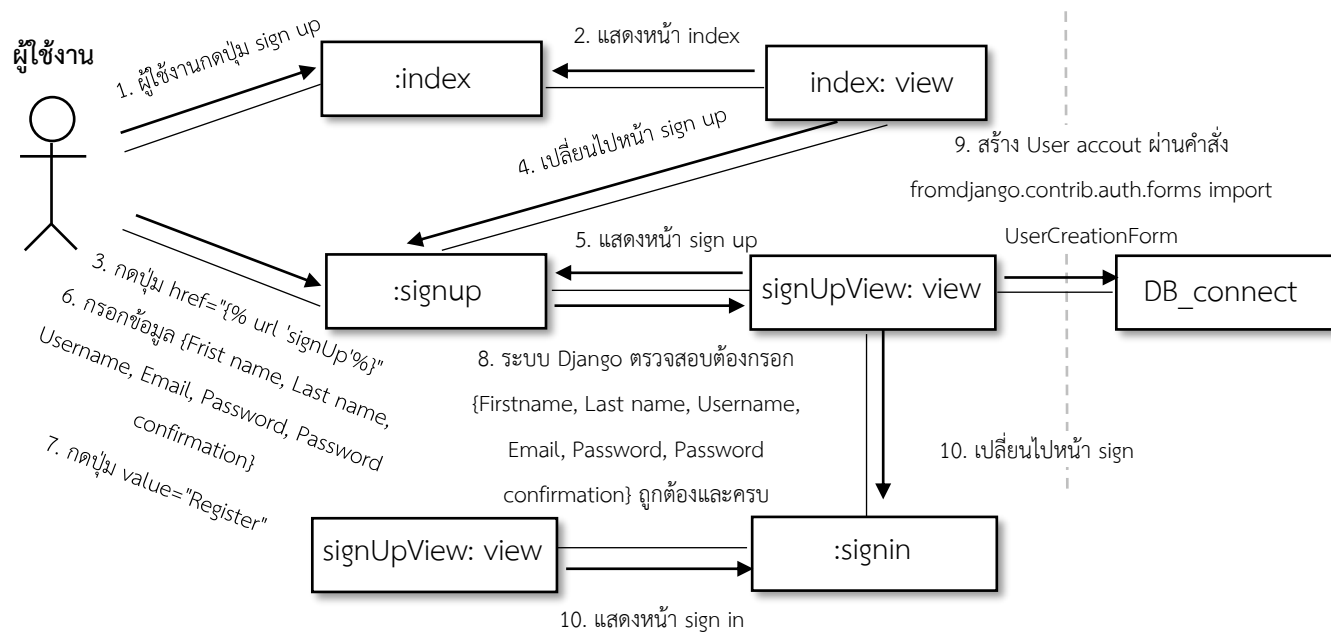
	<p>E3.1. กรอก Password ที่เป็นข้อมูลระบุตัวตน, น้อยกว่า 8 ตัวอักษร, ตัวเลขทั้งหมดและที่นิยมใช้กันทั่วไป</p> <p>E3.2. กดปุ่ม Register</p>	<p>E3.3. ระบบ Django ตรวจสอบต้องกรอก {Firstname, Last name, Username, Email, Password, Password confirmation} ถูกต้องและครบ</p> <p>E3.4. ตรวจสอบผ่านคำสั่ง from django.contrib.auth.forms import UserCreationForm</p> <p>E3.5. ไม่สามารถลงทะเบียนได้</p> <p>E3.6. ระบบแสดงข้อความ alert “This password is too short, It must contain at least 8 characters. , This password is too common.”</p>
--	--	---

ตารางที่ 3-4 Use Case Description: sign up ของ ผู้ใช้งาน

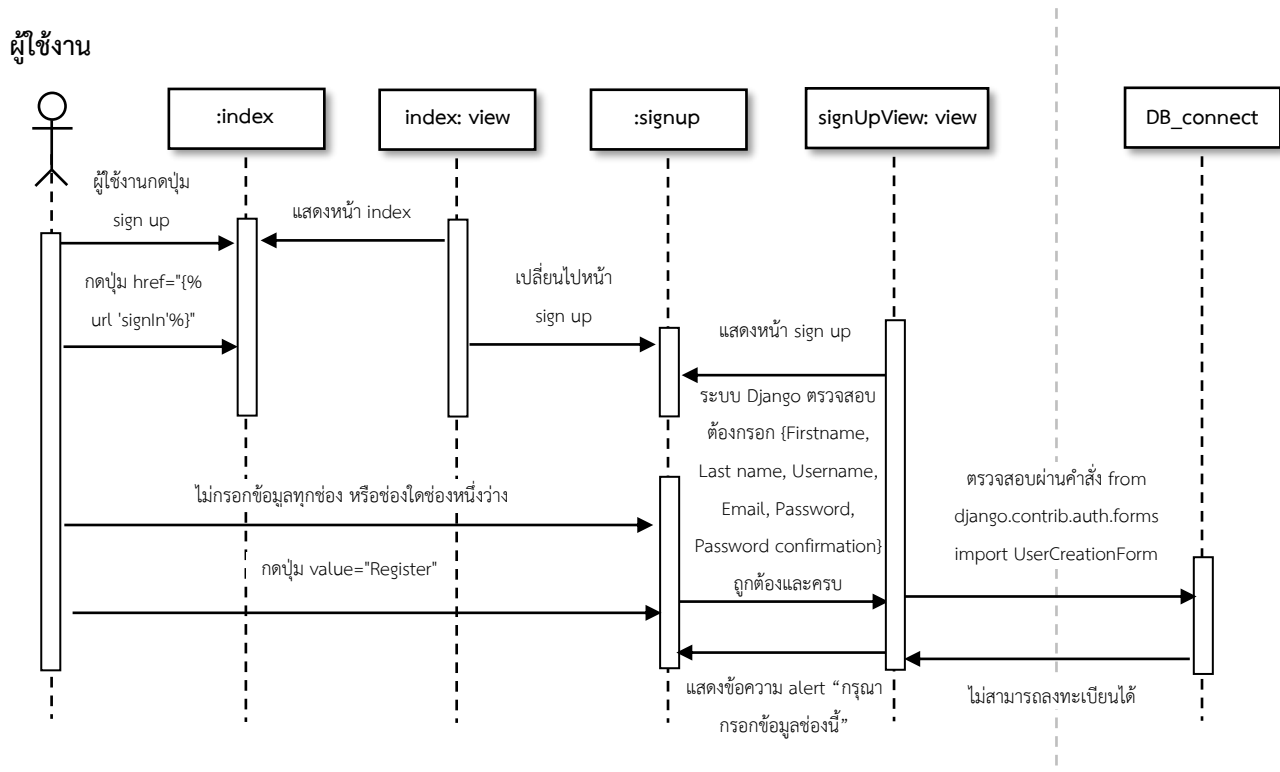
➤ Sequence diagram: sign up ของ ผู้ใช้งาน



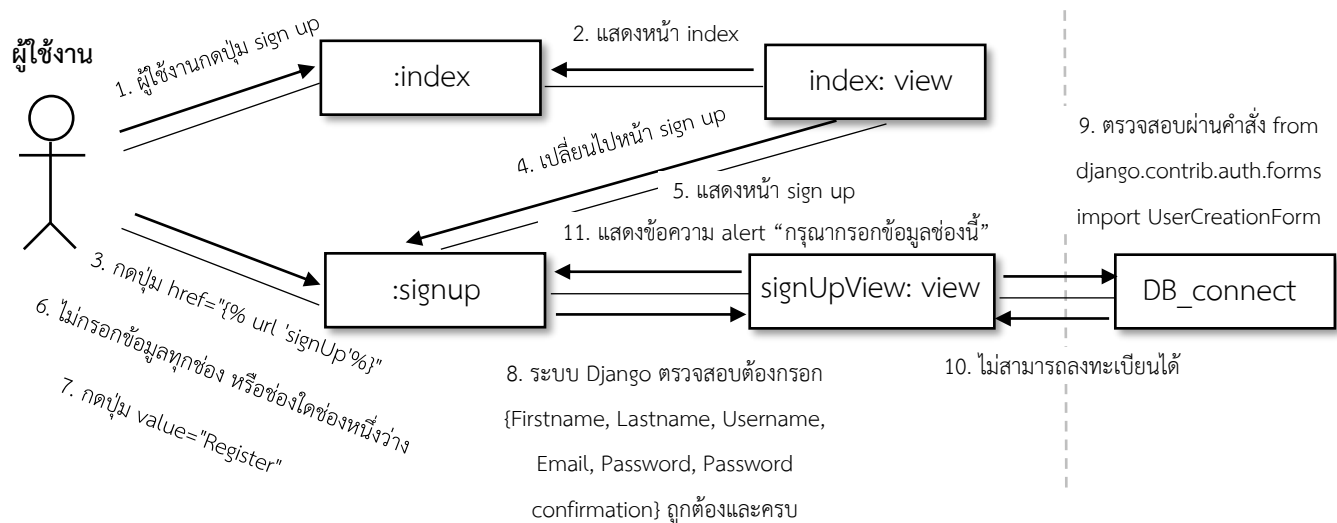
➤ Collaboration Diagram: sign up ของ ผู้ใช้งาน



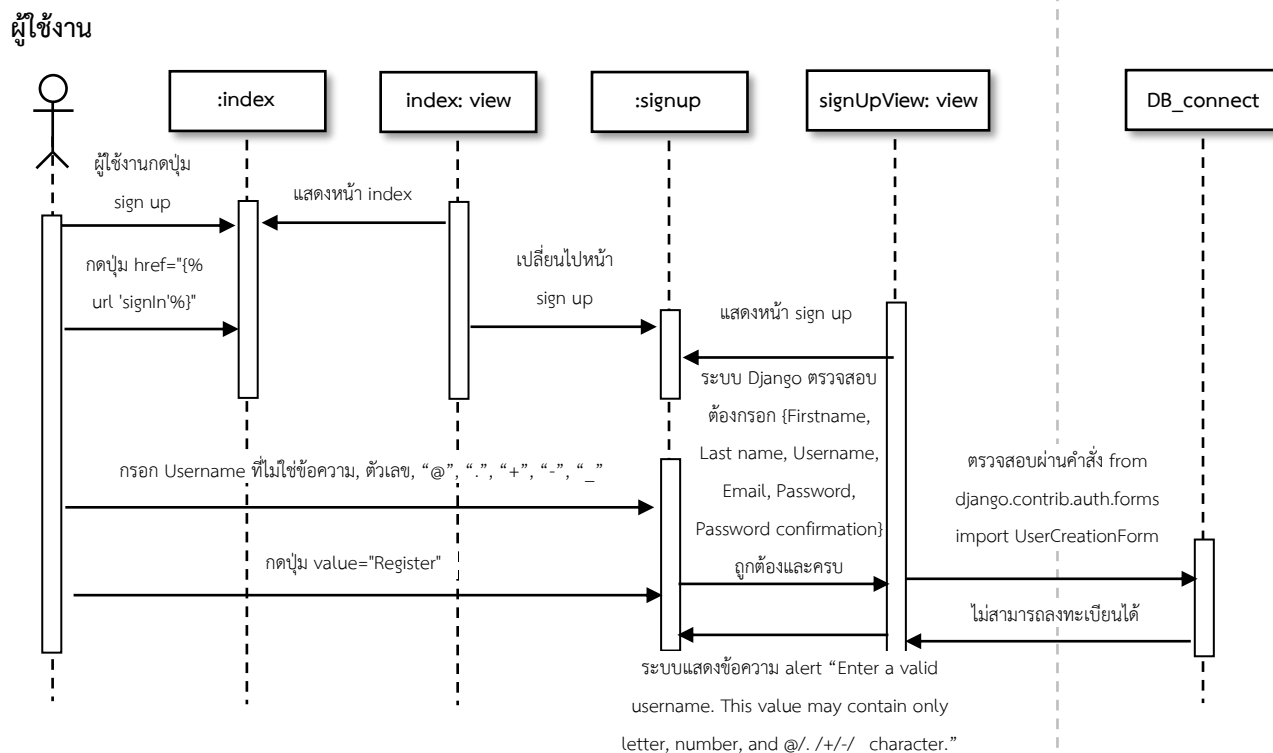
- Sequence diagram: sign up ของ ผู้ใช้งาน กรณี ไม่กรอกข้อมูลทุกช่อง หรือช่องใดช่องหนึ่งว่าง



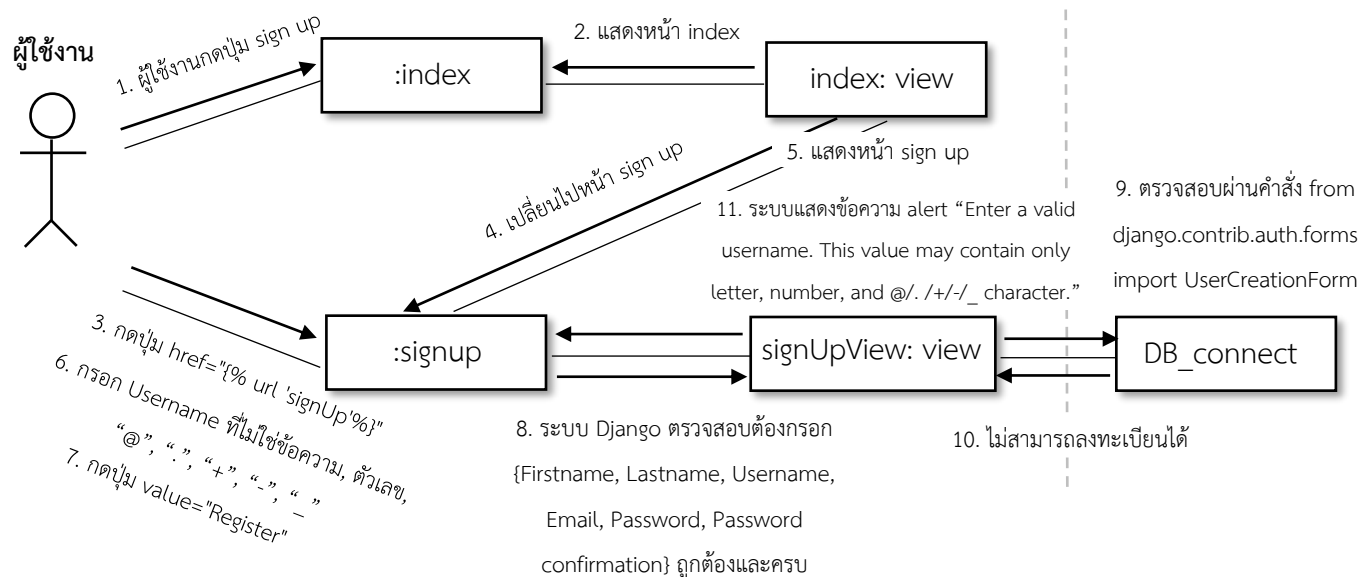
- Collaboration Diagram: sign up ของ ผู้ใช้งาน กรณี ไม่กรอกข้อมูลทุกช่อง หรือช่องใดช่องหนึ่งว่าง



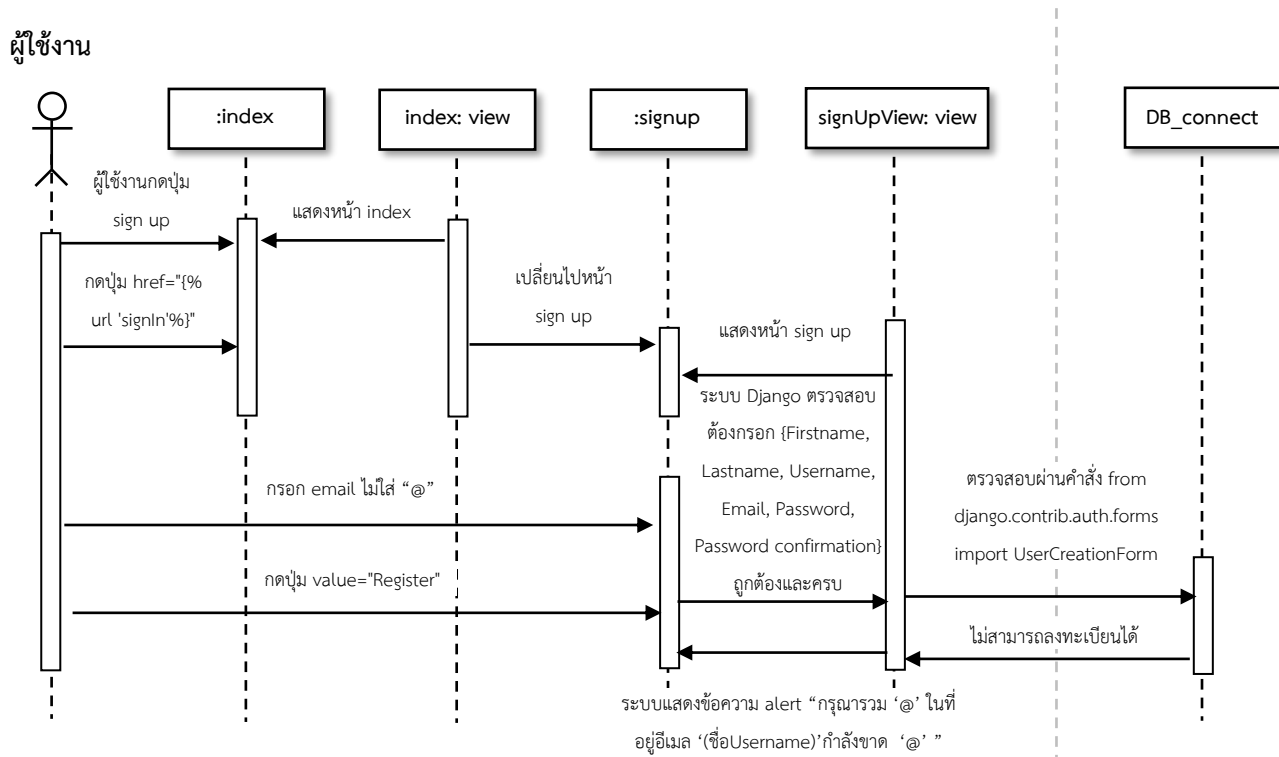
- **Sequence diagram:** sign up ของ ผู้ใช้งาน กรณี กรอก Username ที่ไม่ใช่ข้อความ, ตัวเลข, “@”, “.”, “+”, “-”, “_”



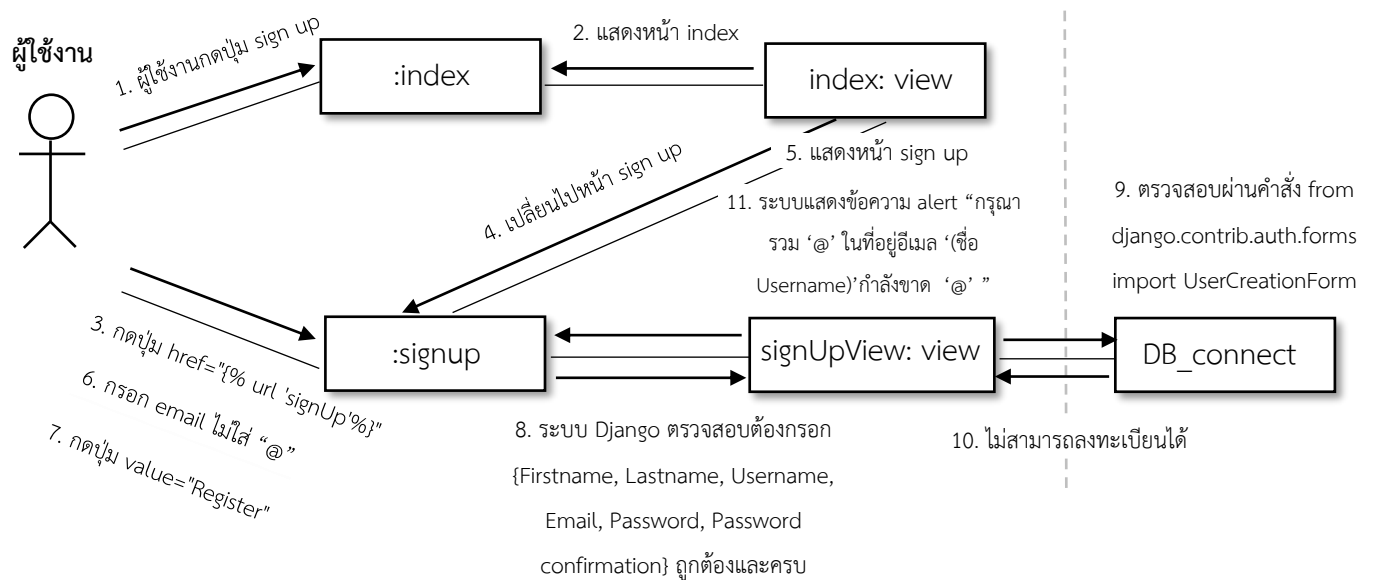
- **Collaboration Diagram:** sign up ของ ผู้ใช้งาน กรณี กรอก Username ที่ไม่ใช่ข้อความ, ตัวเลข, “@”, “.”, “+”, “-”, “_”



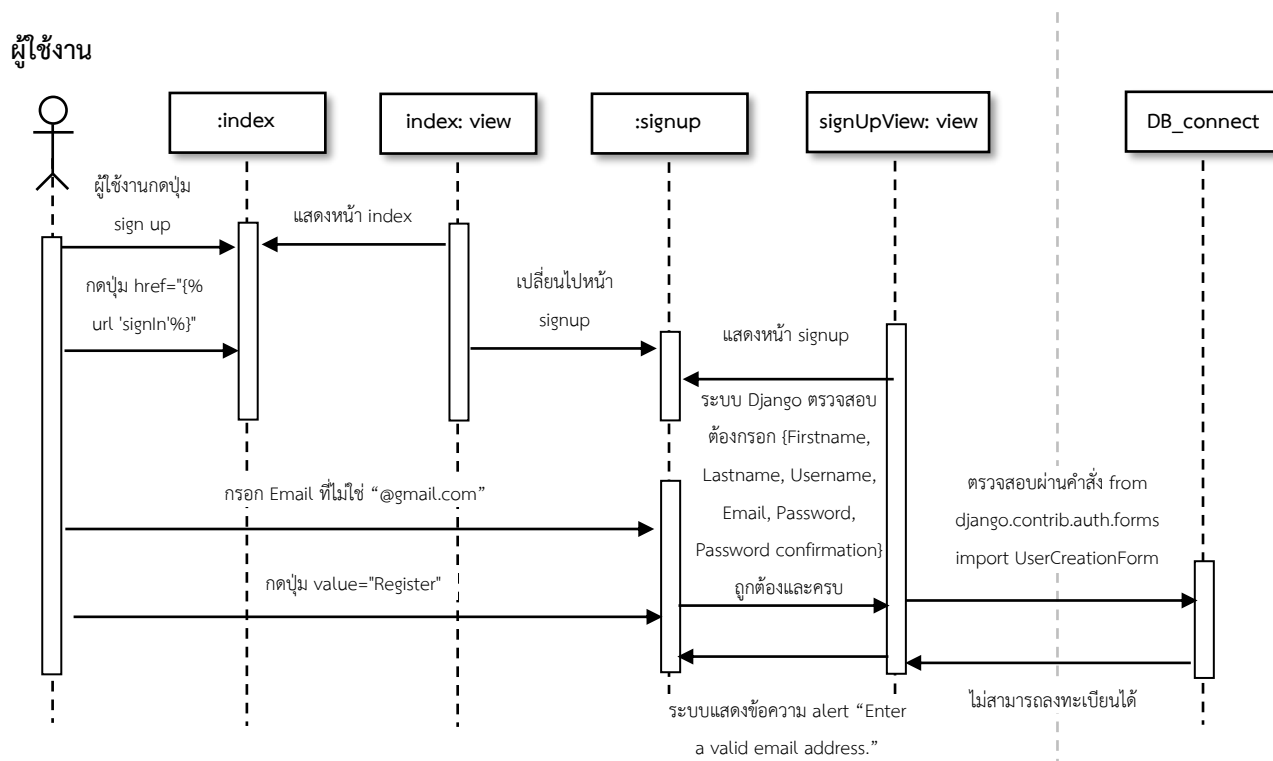
➤ Sequence diagram: sign up ของ ผู้ใช้งาน กรณี กรอก email ไม่ใช่ “@”



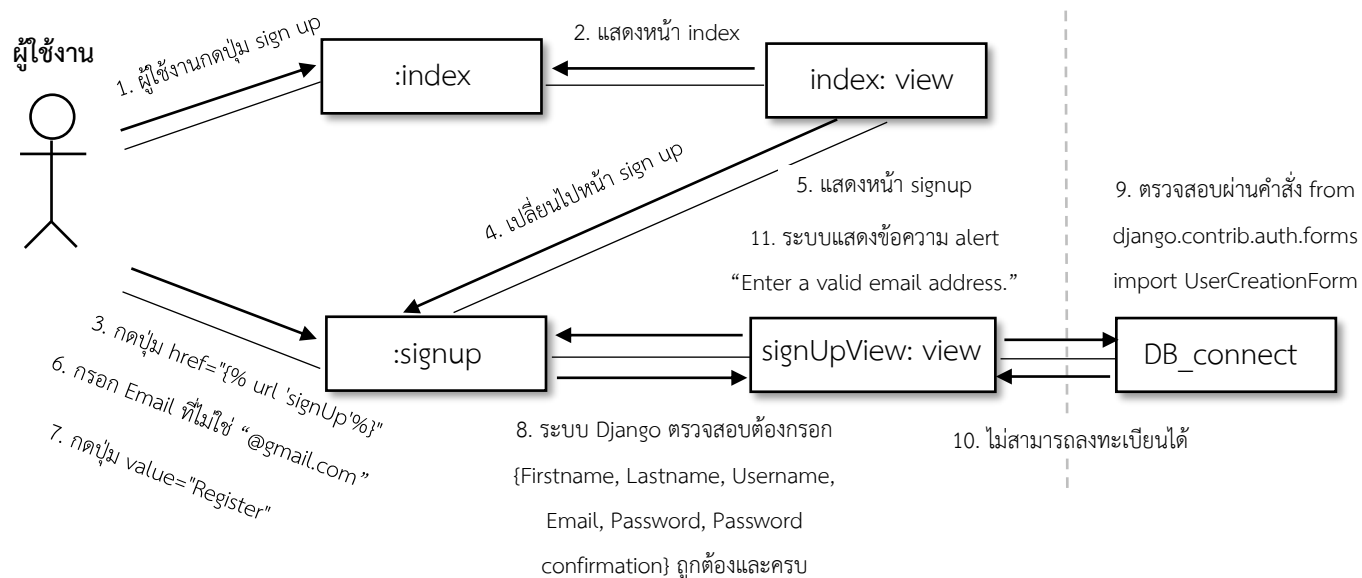
➤ Collaboration Diagram: sign up ของ ผู้ใช้งาน กรณี กรอก email ไม่ใช่ “@”



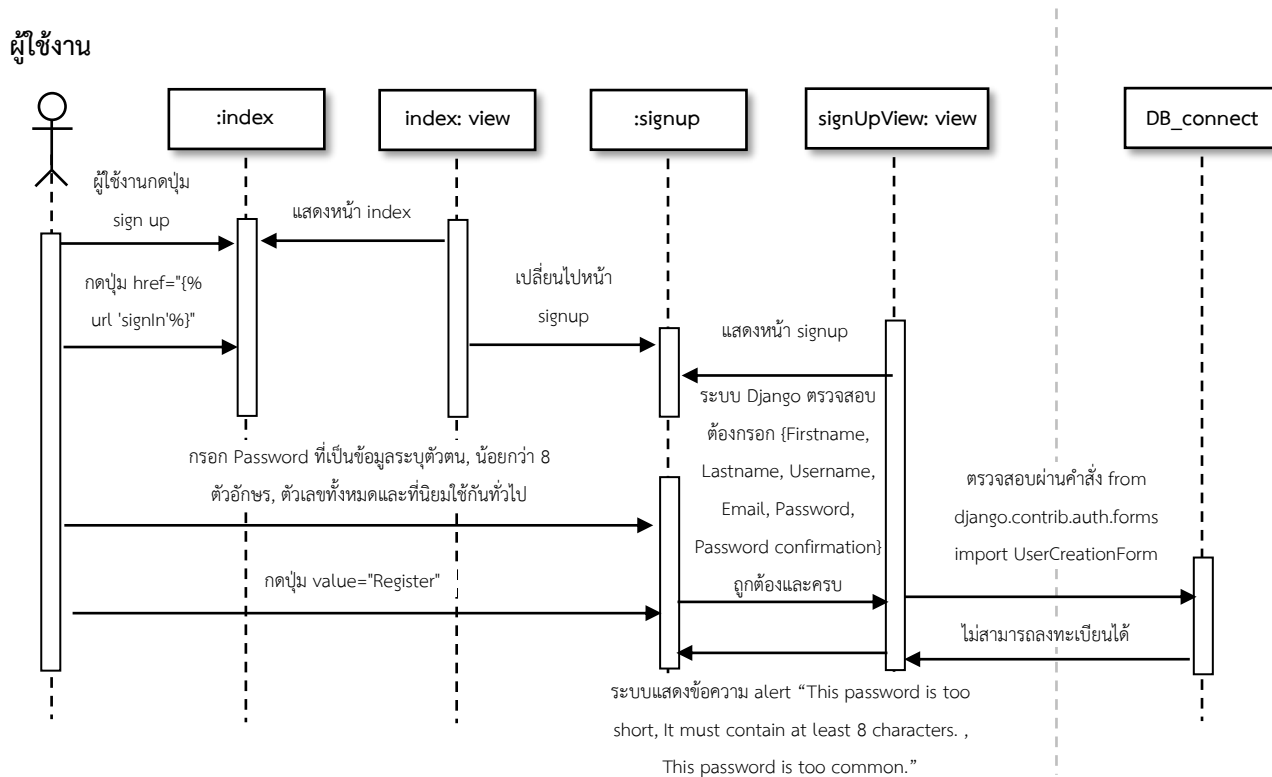
➤ Sequence diagram: sign up ของ ผู้ใช้งาน กรณี กรอก Email ที่ไม่ใช่ “@gmail.com”



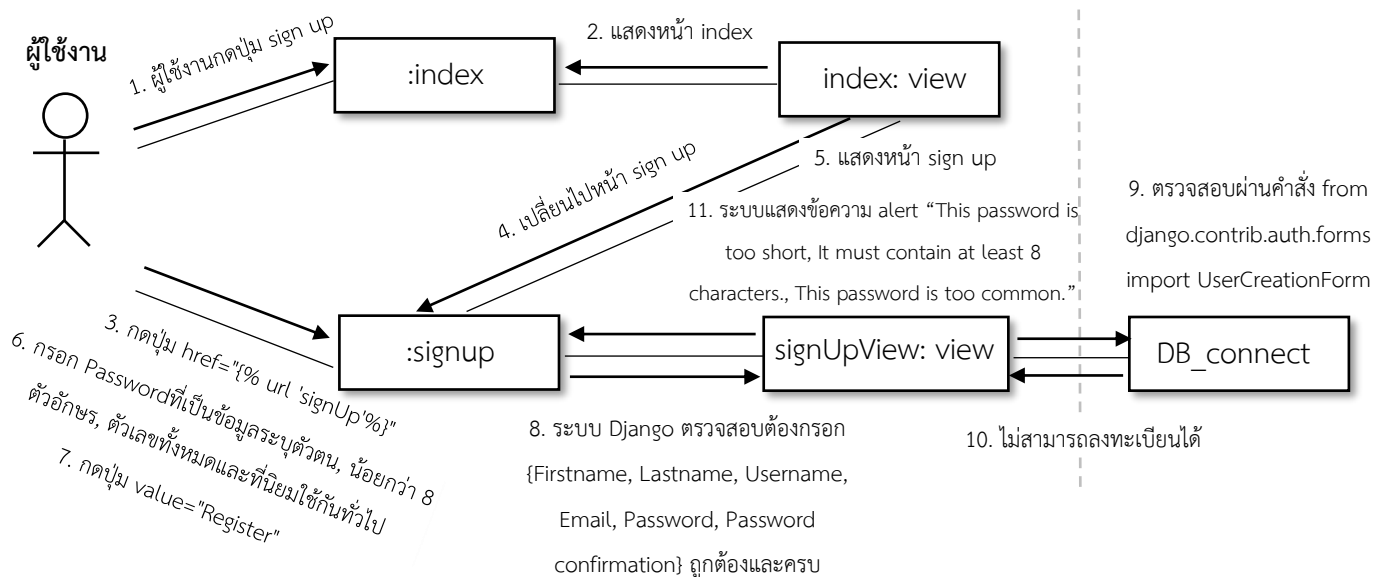
➤ Collaboration Diagram: sign up ของ ผู้ใช้งาน กรณี กรอก Email ที่ไม่ใช่ “@gmail.com”



- **Sequence diagram:** sign up ของ ผู้ใช้งาน กรณี กรอก Password ที่เป็นข้อมูลระบุตัวตน, น้อยกว่า 8 ตัวอักษร, ตัวเลขทั้งหมดและที่นิยมใช้กันทั่วไป



- **Collaboration Diagram:** sign up ของ ผู้ใช้งาน กรณี กรอก Password ที่เป็นข้อมูลระบุตัวตน, น้อยกว่า 8 ตัวอักษร, ตัวเลขทั้งหมดและที่นิยมใช้กันทั่วไป



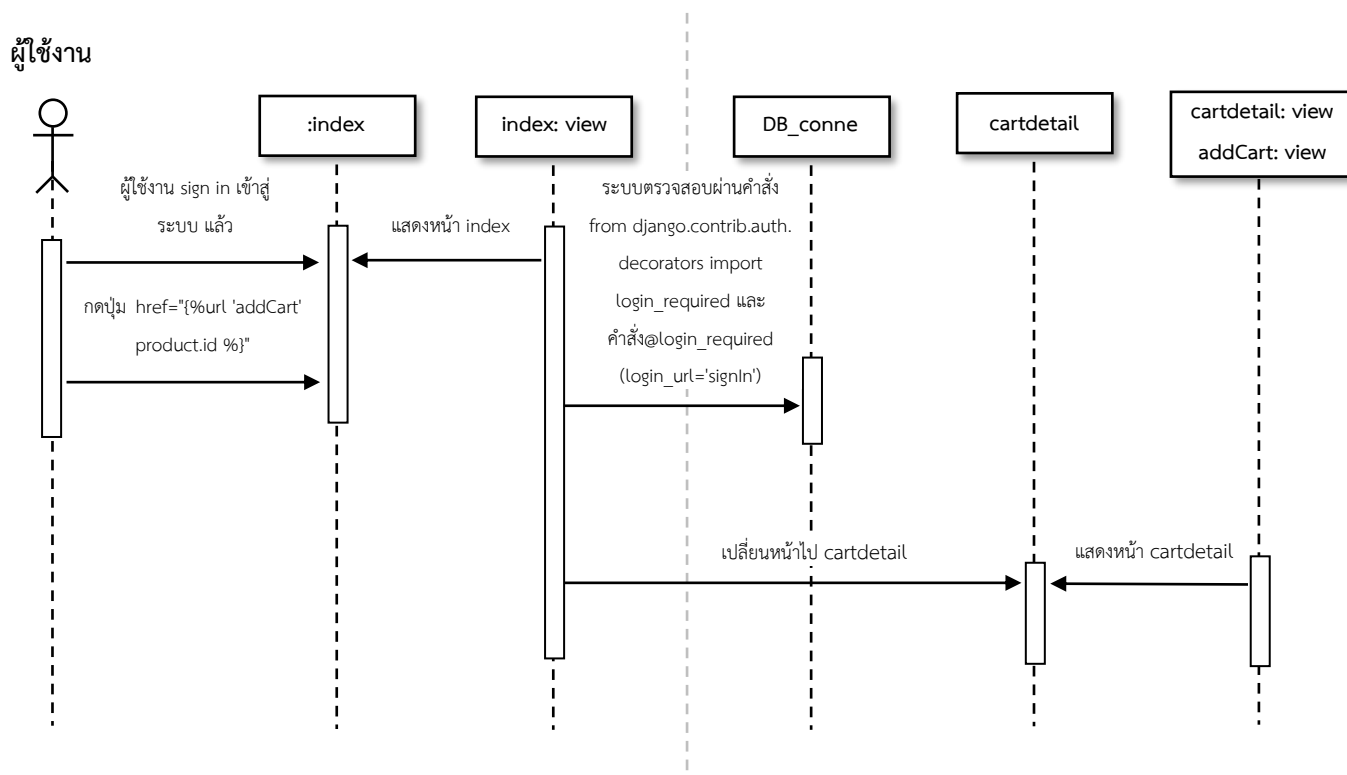
➤ Use Case Description: ผู้ใช้เพิ่มสินค้าลงตะกร้า

Use case	4	
Use case name	เพิ่มสินค้าลงตะกร้า	
Actor	ผู้ใช้งาน	
Description	เลือกสินค้าภายในร้านลงตะกร้า	
Precondition	เลือกสินค้าภายในร้านหากผู้ใช้ได้ทำการ sign in เข้ามาแล้ว	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่มเพิ่มสินค้าลงตะกร้า	2. ระบบตรวจสอบผ่านคำสั่ง from django.contrib.auth. decorators import login_required และ คำสั่ง@login_required (login_url='signIn') 3. แสดงหน้าตะกร้าสินค้าที่มี สินค้าอยู่ภายใน
Post condition	สามารถเพิ่มสินค้าลงตะกร้าได้	
Alternative Course 1	Actor Action	System Response
	A1.1. ผู้ใช้ยังไม่ได้ทำการ sign in เข้าสู่ระบบแต่กดเพิ่มสินค้าลงตะกร้า	A1.2. ระบบตรวจสอบผ่านคำสั่ง from django.contrib.auth. decorators import login_required และคำสั่ง @login_required (login_url='signIn') A1.3. ไม่สามารถเพิ่มสินค้าลง ตะกร้าได้

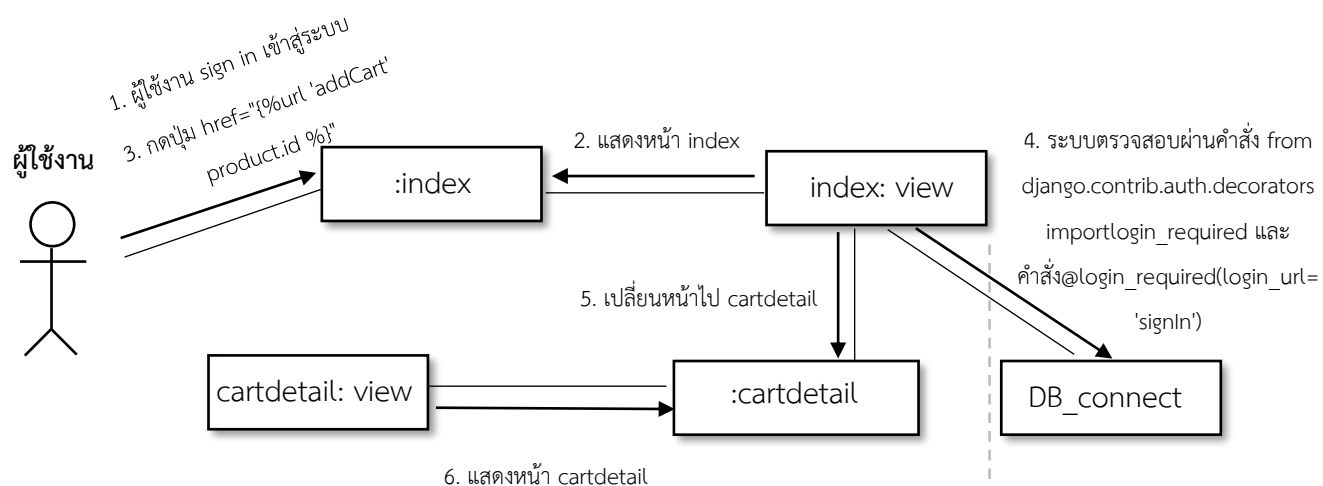
		A1.4. ระบบแสดงหน้า sign in
Alternative Course 2	Actor Action	System Response
	B1.1. ยังไม่มีสินค้าในตะกร้าแต่กดเข้าหน้าตะกร้าสินค้า	<p>B1.2. ระบบตรวจสอบผ่านคำสั่ง from django.contrib.auth.decorators import login_required และคำสั่ง @login_required (login_url='signIn')</p> <p>B1.3. ระบบแสดงข้อความ alert “ไม่มีสินค้าในตะกร้า”</p>

ตารางที่ 3-5 Use Case Description: ผู้ใช้เพิ่มสินค้าลงตะกร้า

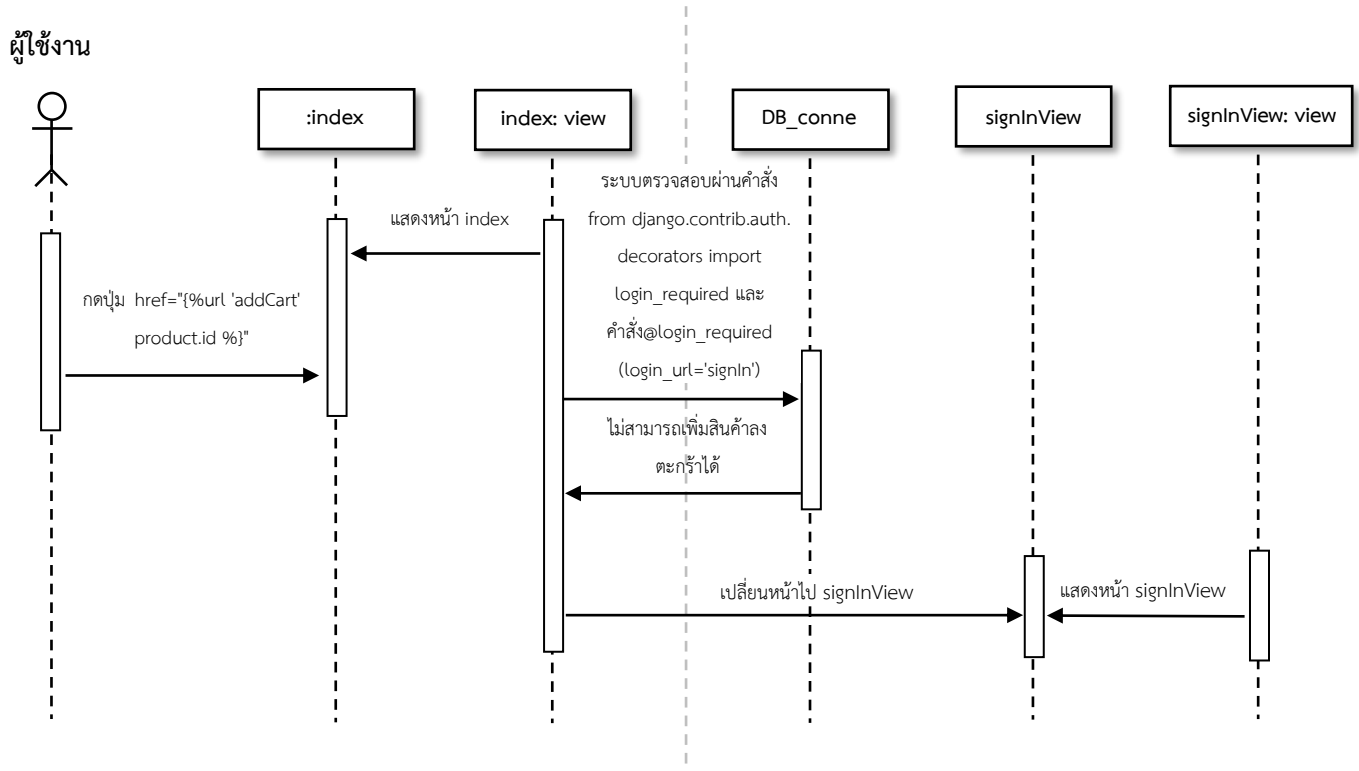
➤ Sequence diagram: ผู้ใช้เพิ่มสินค้าลงตะกร้า



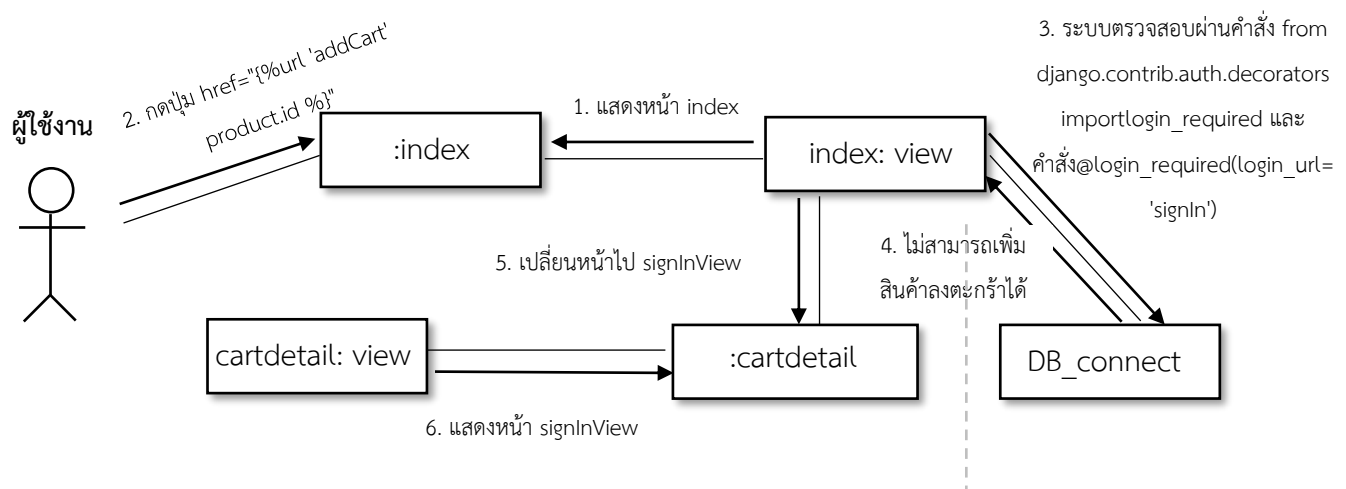
➤ Collaboration Diagram: ผู้ใช้เพิ่มสินค้าลงตะกร้า



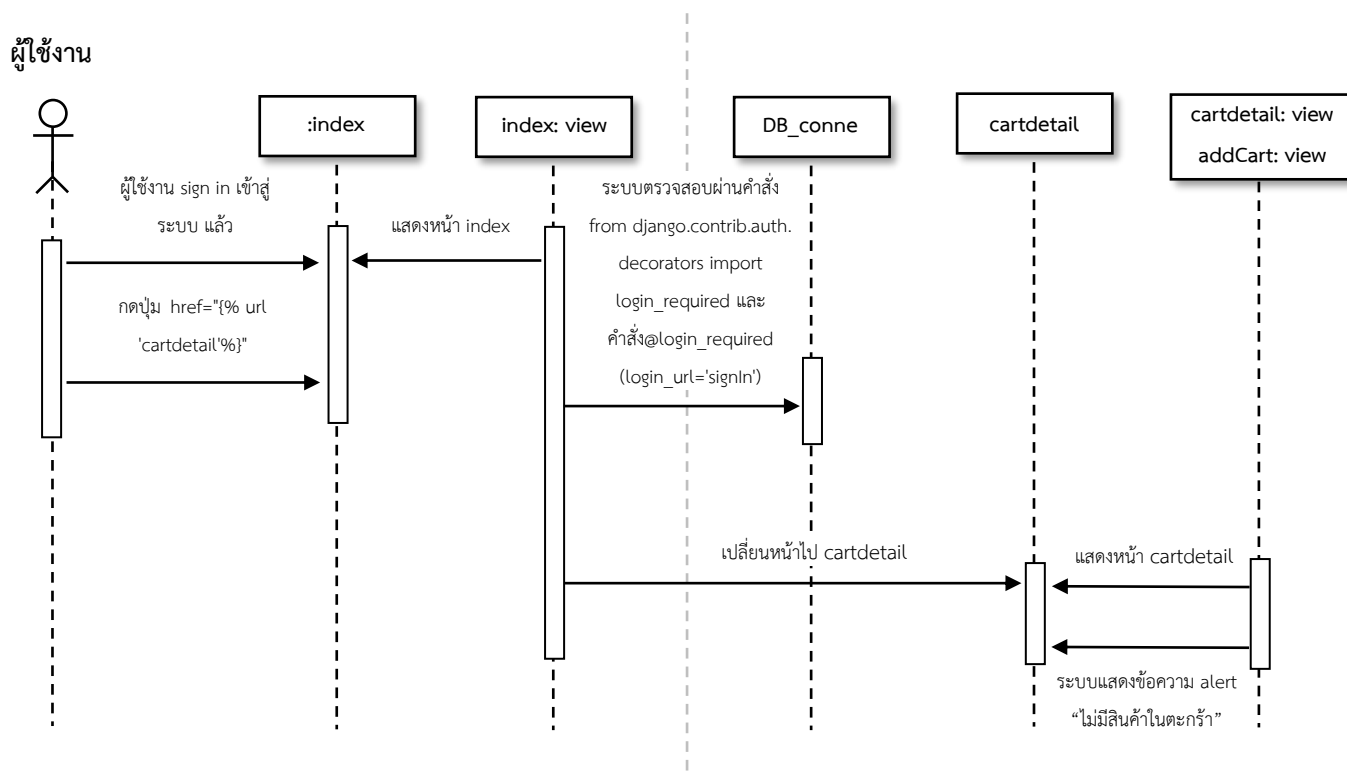
- **Sequence diagram:** ผู้ใช้เพิ่มสินค้าลงตะกร้า กรณี ผู้ใช้ยังไม่ได้ทำการ sign in เข้าสู่ระบบแต่กดเพิ่มสินค้าลงตะกร้า



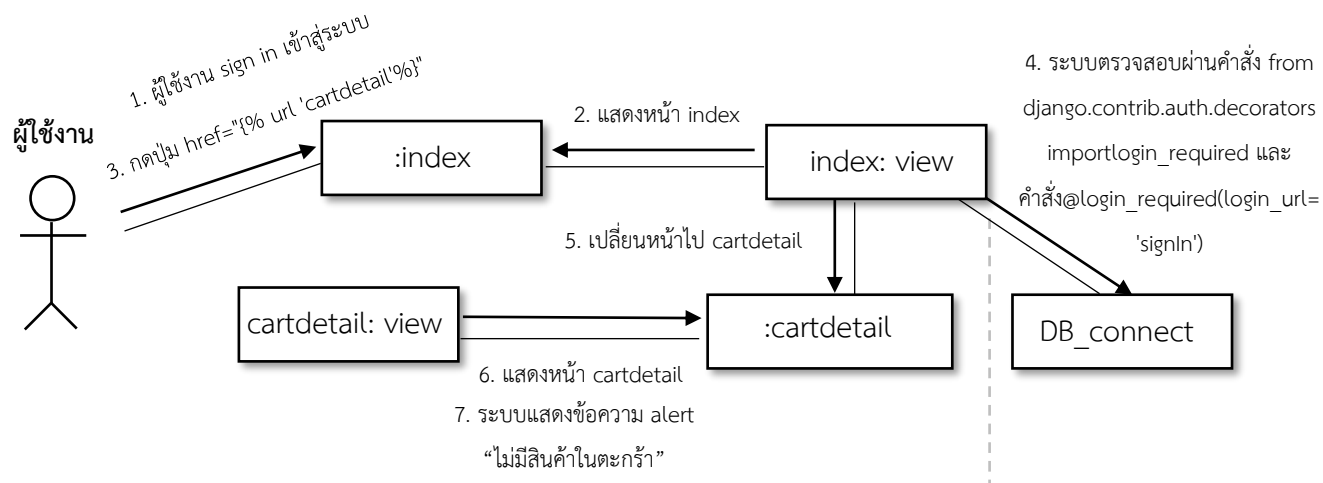
- **Collaboration Diagram:** ผู้ใช้เพิ่มสินค้าลงตะกร้า กรณี ผู้ใช้ยังไม่ได้ทำการ sign in เข้าสู่ระบบแต่กดเพิ่มสินค้าลงตะกร้า



- **Sequence diagram:** ผู้ใช้เพิ่มสินค้าลงตะกร้า กรณี ยังไม่มีสินค้าในตะกร้าแต่กดเข้าหน้าตะกร้าสินค้า



- **Collaboration Diagram:** ผู้ใช้เพิ่มสินค้าลงตะกร้า กรณี ยังไม่มีสินค้าในตะกร้าแต่กดเข้าหน้าตะกร้าสินค้า

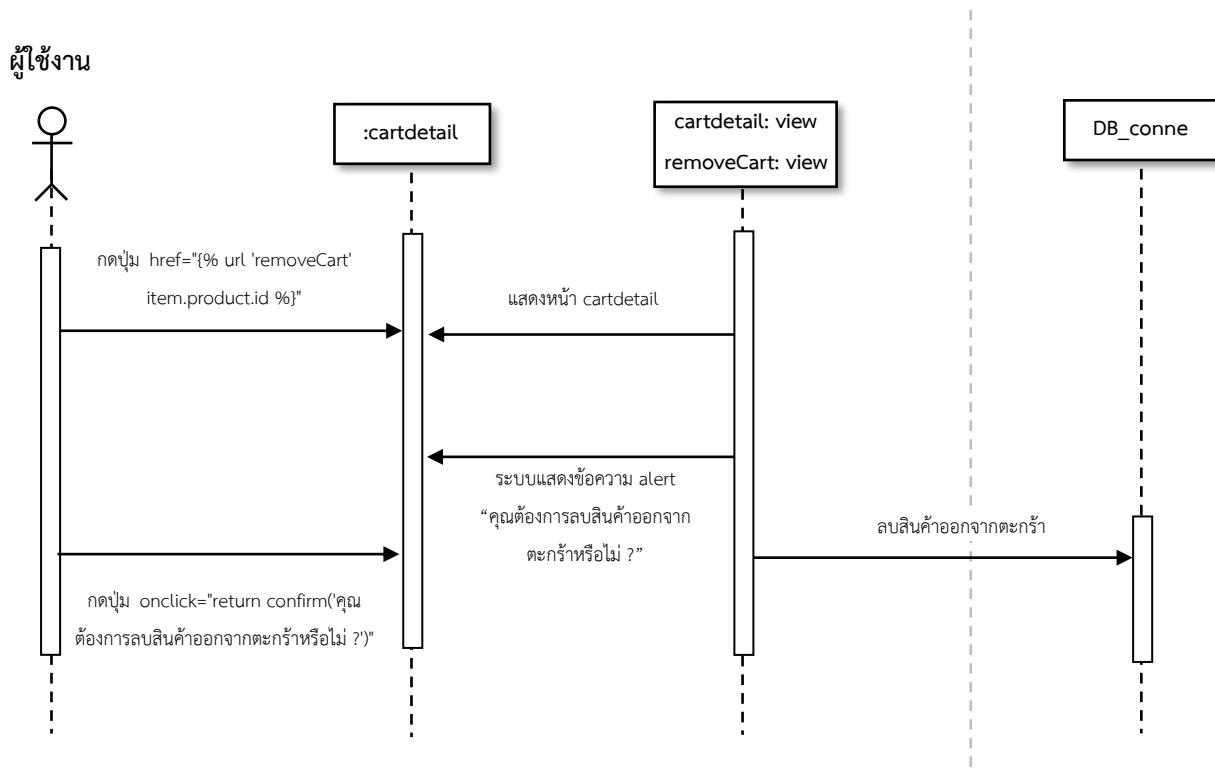


➤ Use Case Description: ผู้ใช้ลบสินค้าออกจากตะกร้า

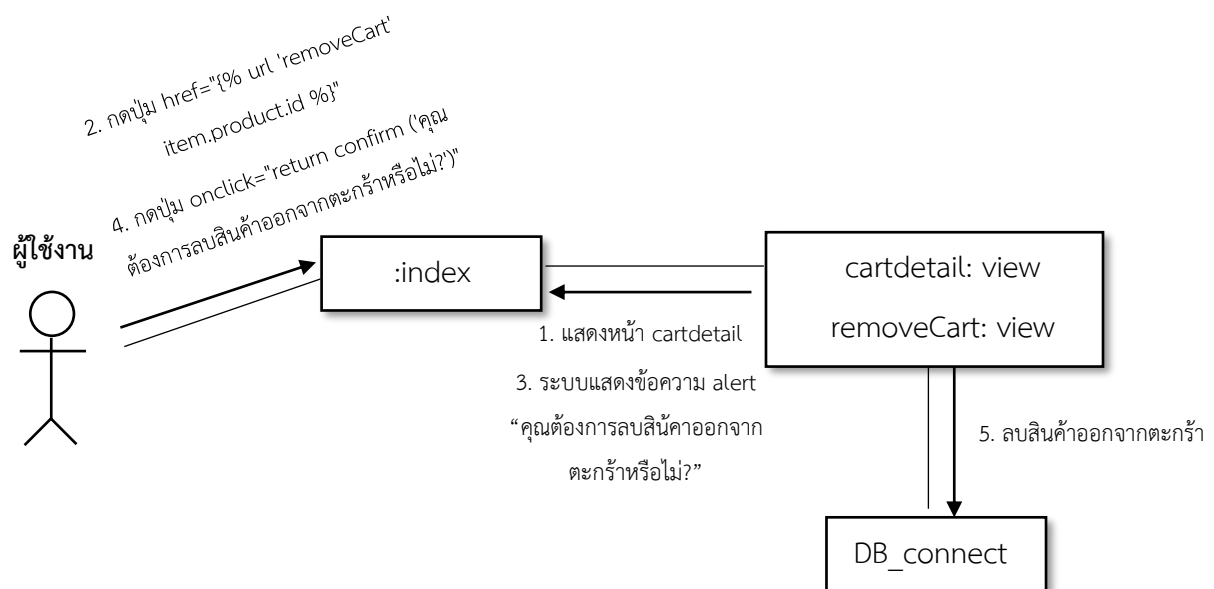
Use case	5	
Use case name	ลบสินค้าออกจากตะกร้า	
Actor	ผู้ใช้งาน	
Description	เอารายการสินค้าที่ไม่ต้องการแล้วออกจากตะกร้า	
Precondition	มีสินค้าอยู่ในตะกร้า	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่ม X หรือปุ่มลบรายการสินค้าออกจากตะกร้า 3. ผู้ใช้กดปุ่ม “ตกลง”	2. ระบบแสดงข้อความ alert “คุณต้องการลบสินค้าออกจากตะกร้าหรือไม่ ?” 4. ระบบลบสินค้าออกจากตะกร้า 5. แสดงหน้าตะกร้าสินค้าและรายการสินค้าปัจจุบัน
Post condition	สามารถลบสินค้าในตะกร้าได้	
Alternative Course 1	Actor Action	System Response
	A3.1. ลบรายการสินค้ารายการสุดท้ายในตะกร้า A3.3. ผู้ใช้กดปุ่ม “ตกลง”	A3.2. ระบบแสดงแจ้งเตือน “คุณต้องการลบสินค้าออกจากตะกร้าหรือไม่ ?” A3.4. ระบบแสดงหน้าตะกร้าสินค้าและแสดงข้อความ alert “ไม่มีสินค้าในตะกร้า”

ตารางที่ 3-6 Use Case Description: ผู้ใช้ลบสินค้าออกจากตะกร้า

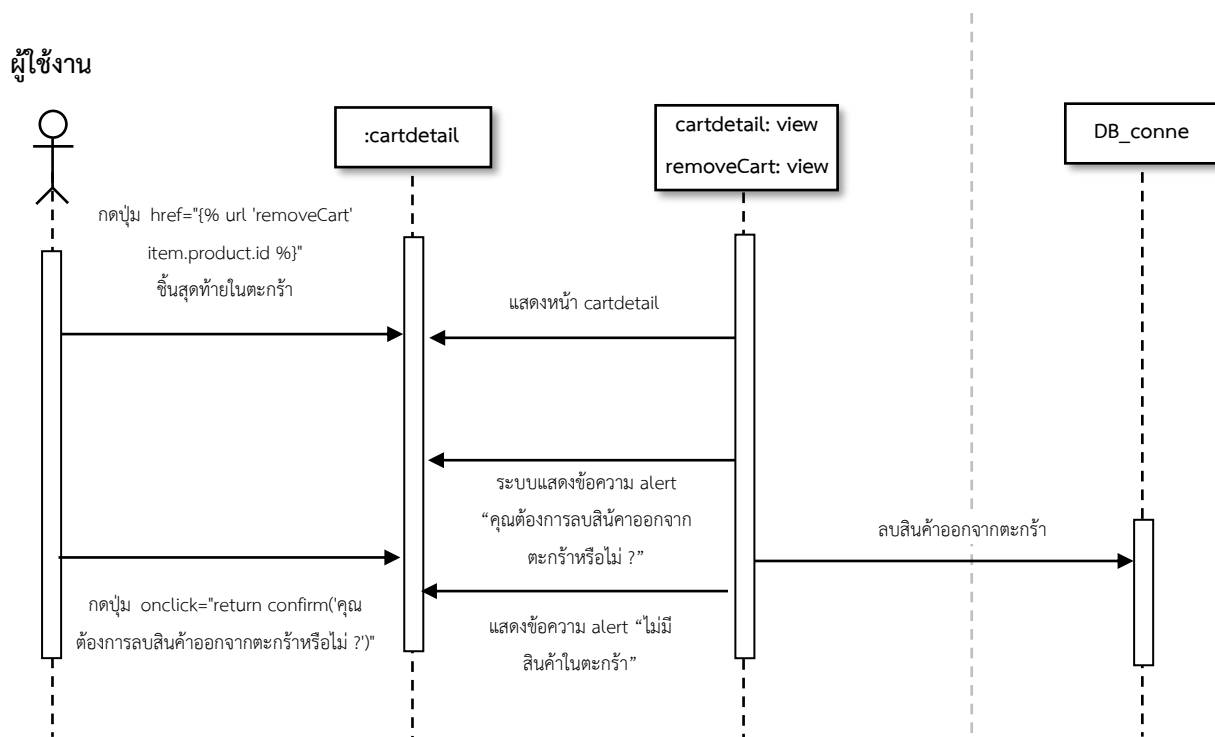
➤ Sequence diagram: ผู้ใช้ลบสินค้าออกจากตะกร้า



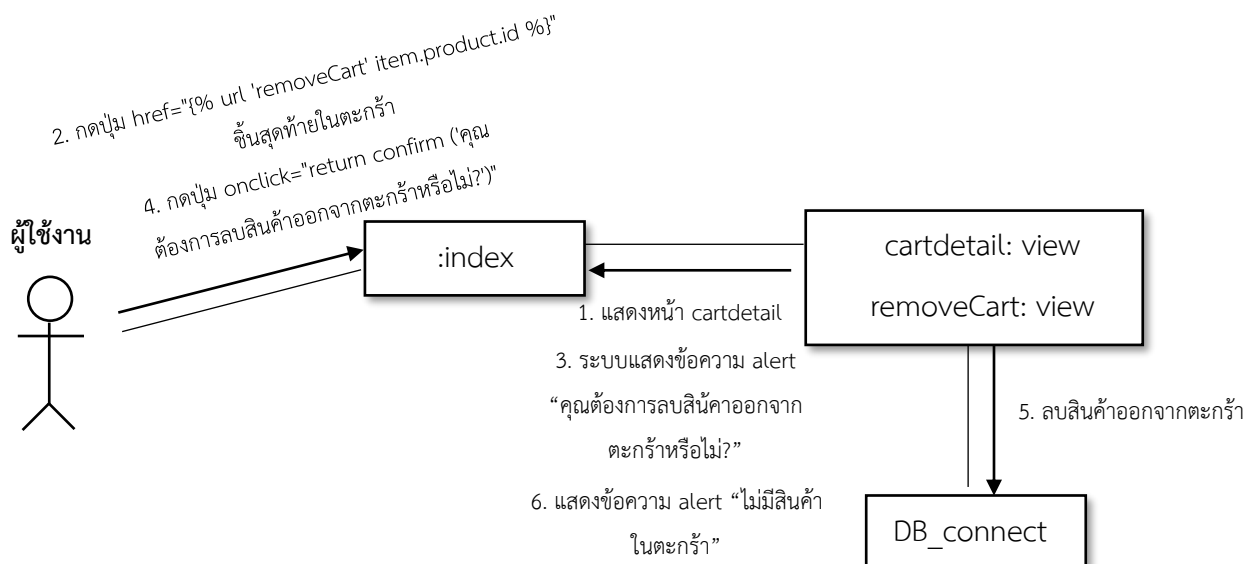
➤ Collaboration Diagram: ผู้ใช้ลบสินค้าออกจากตะกร้า



- Sequence diagram: ผู้ใช้ลบสินค้าออกจากตะกร้า กรณี ลบรายการสินค้ารายการสุดท้ายในตะกร้า



- Collaboration Diagram: ผู้ใช้ลบสินค้าออกจากตะกร้า กรณี ลบรายการสินค้ารายการสุดท้ายในตะกร้า

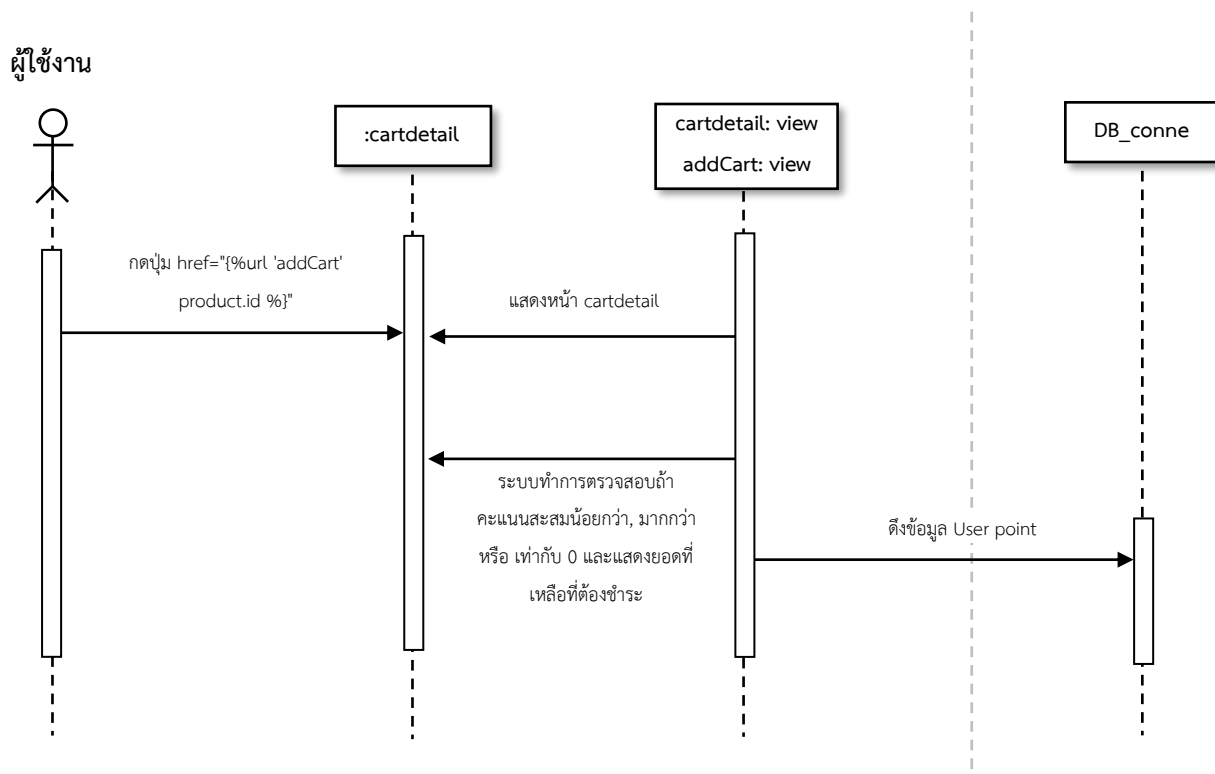


➤ Use Case Description: แต้มคะแนนสะสม

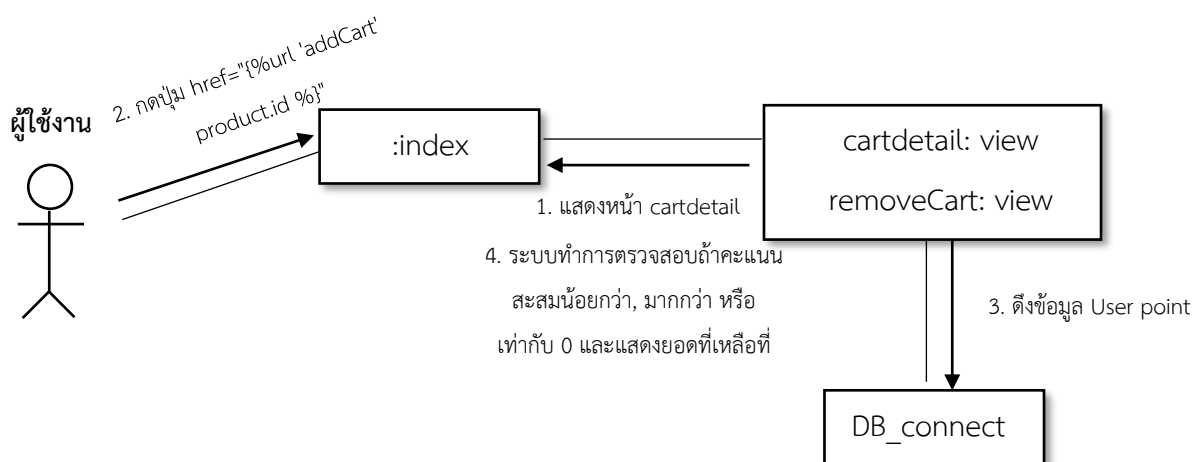
Use case	6	
Use case name	ได้ส่วนลดจากแต้มคะแนนสะสม	
Actor	ผู้ใช้งาน	
Description	ได้ส่วนลดจากแต้มคะแนนสะสมในการสั่งซื้อสินค้า	
Precondition	ผู้ใช้งาน Sing in เข้าสู่ระบบ	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่มเพิ่มสินค้าลงตะกร้า	2. ดึงข้อมูล User point 3.1. ระบบทำการตรวจสอบถ้าคะแนนสะสมน้อยกว่ายอดรวมทั้งหมด จะแสดงยอดที่เหลือที่ต้องชำระ 3.2. ระบบทำการตรวจสอบถ้าคะแนนสะสมมากกว่ายอดรวมทั้งหมด จะแสดงยอดที่ต้องชำระเท่ากับ 0 3.3. ระบบทำการตรวจสอบถ้ามีแต้มคะแนนสะสมเป็น 0 จะแสดงยอดที่ต้องชำระ 4. ระบบแสดงคะแนนที่เหลือบนแถบนำทางหรือนavigator bar และแสดงหน้าตะกร้าสินค้าและยอดหลังหักส่วนลด
Post condition	หน้าตะกร้าสินค้าที่แสดงรายการสินค้าและยอดที่ต้องชำระหลังหักส่วนลด	

ตารางที่ 3-7 Use Case Description: แต้มคะแนนสะสม

➤ Sequence diagram: แต้มคะแนนสะสม



➤ Collaboration Diagram: แต้มคะแนนสะสม



➤ Use Case Description: ผู้ใช้ชำระเงิน

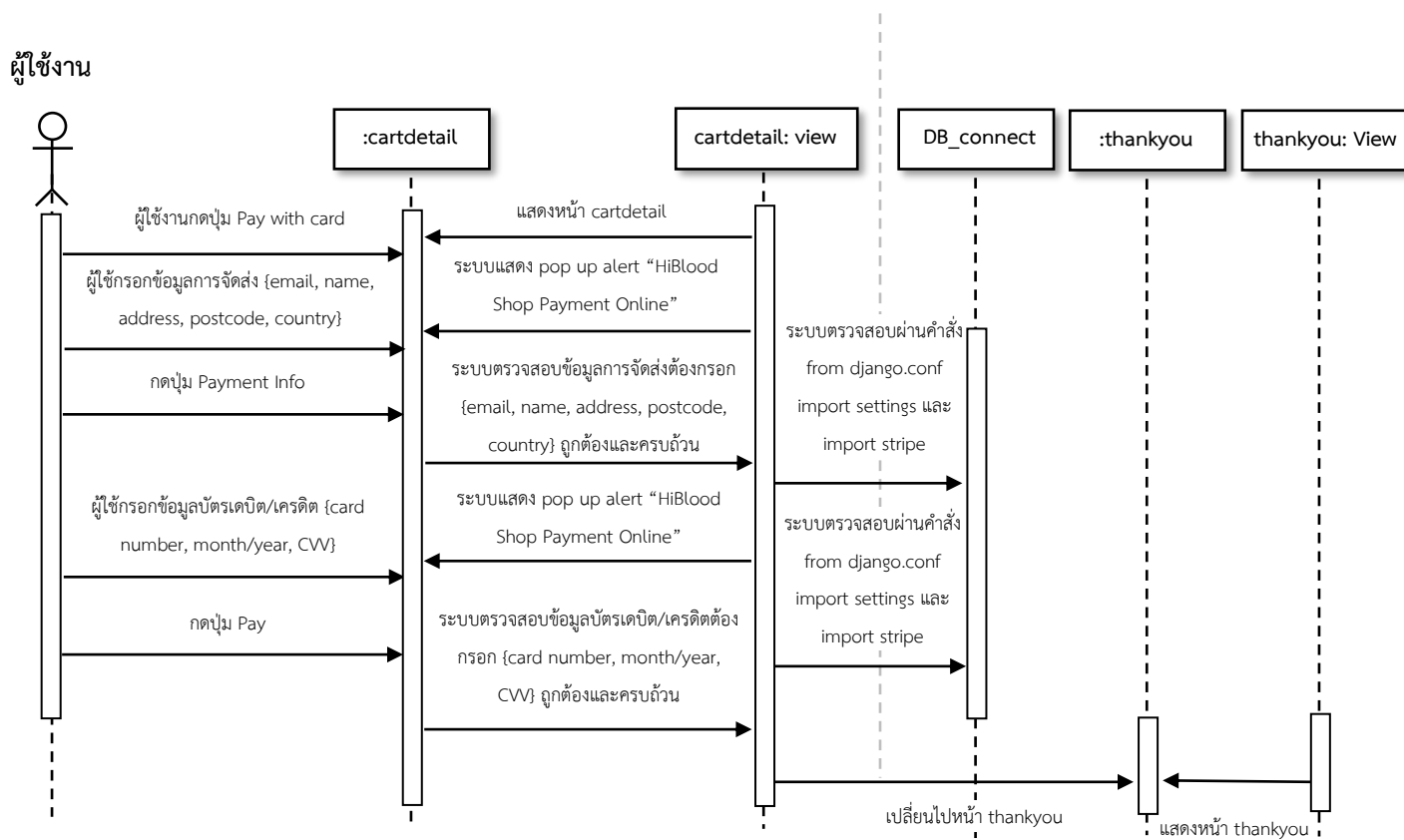
Use case	7	
Use case name	ชำระเงิน	
Actor	ผู้ใช้งาน	
Description	ผู้ใช้ทำการชำระเงินหลังจากเลือกสินค้าลงตะกร้าเรียบร้อยแล้ว	
Precondition	มีสินค้าอยู่ในตะกร้า	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานกดปุ่ม Pay with card 3. ผู้ใช้กรอกข้อมูลการจัดส่ง {email, name, address, postcode, country} 4. ผู้ใช้กดปุ่ม Payment Info 8. ผู้ใช้กรอกข้อมูลบัตรเครดิต/เครดิต {card number, month/year, CVV}	2. ระบบแสดงป๊อปอัพ “HiBlood Shop Payment Online” 5. ระบบตรวจสอบข้อมูลการจัดส่งต้องกรอก {email, name, address, postcode, country} ถูกต้องและครบถ้วน 6.ระบบตรวจสอบผ่านคำสั่ง from django.conf import settings และ import stripe 7. ระบบแสดงป๊อปอัพ “HiBlood Shop Payment Online”

	9. ผู้ใช้กดปุ่ม Pay	10. ระบบตรวจสอบข้อมูลบัตรเดบิต/เครดิตต้องกรอก {card number, month/year, CVV} ถูกต้องและครบถ้วน 11. ระบบตรวจสอบผ่านคำสั่ง from django.conf import settings และ import stripe 12. เปลี่ยนไปหน้า thankyou
Post condition	ผู้ใช้ชำระเงินได้และเปลี่ยนไปหน้า Thank you!	
Alternative Course 1	Actor Action	System Response
	A3.1. ผู้ใช้ไม่กรอกข้อมูลการจัดส่งทุกช่อง หรือไม่กรอกช่องใดช่องหนึ่ง A3.2. ผู้ใช้กดปุ่ม Payment Info	A3.3. ระบบตรวจสอบข้อมูลการจัดส่งต้องกรอก {email, name, address, postcode, country} ถูกต้องและครบถ้วน A3.4. ไม่สามารถเก็บข้อมูลการจัดส่งได้ A3.5. แสดงหน้าเดิม
Alternative Course 2	Actor Action	System Response
	A8.1. ผู้ใช้ไม่กรอกข้อมูลข้อมูลบัตรเดบิต/เครดิต หรือไม่กรอกช่องใดช่องหนึ่ง A8.2. ผู้ใช้กดปุ่ม Pay	A8.3. ระบบตรวจสอบข้อมูลบัตรเดบิต/เครดิตต้องกรอก {card number, month/year, CVV} ถูกต้องและครบถ้วน A8.4. ไม่สามารถชำระเงินได้ A8.5. แสดงหน้าเดิม

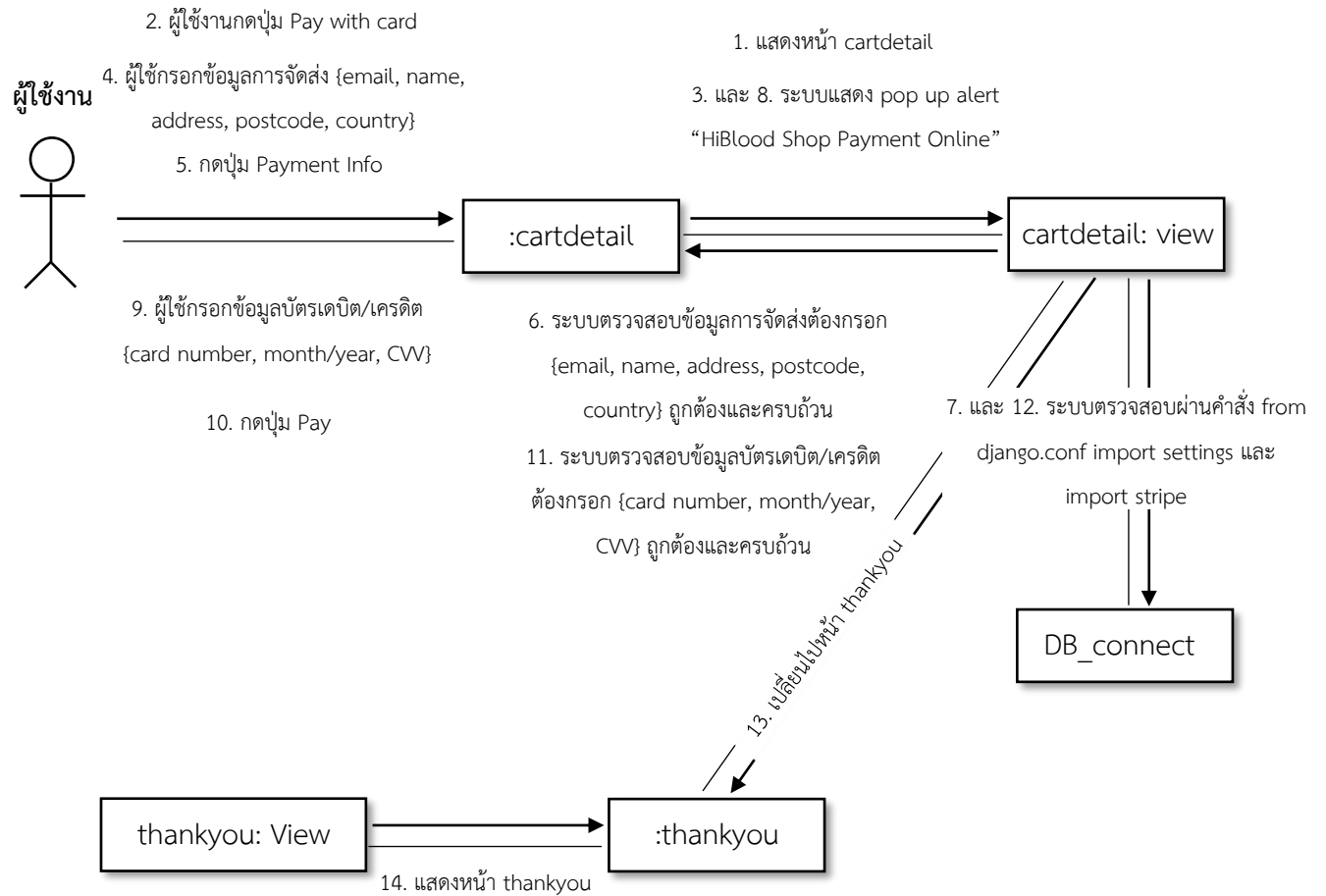
ตารางที่ 3-8 Use Case Description: ผู้ใช้ชำระเงิน

➤ Sequence diagram: ผู้ใช้ชำระเงิน

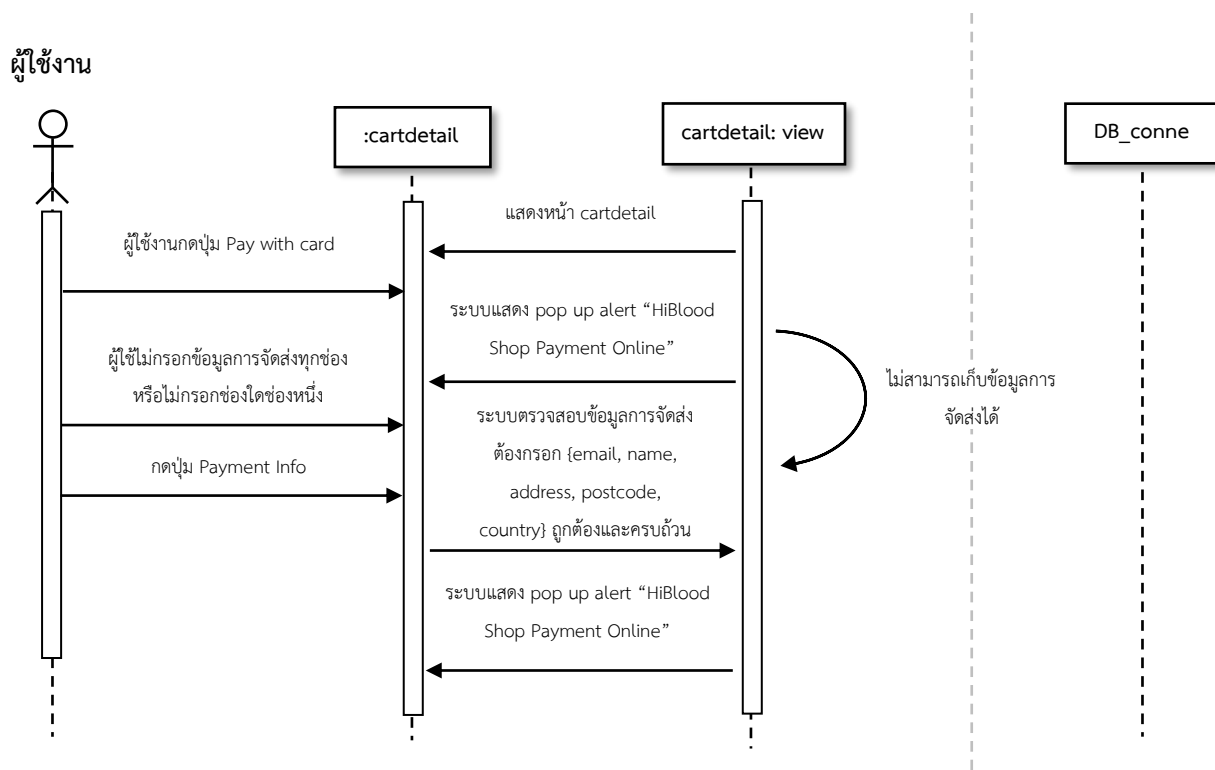
ผู้ใช้งาน



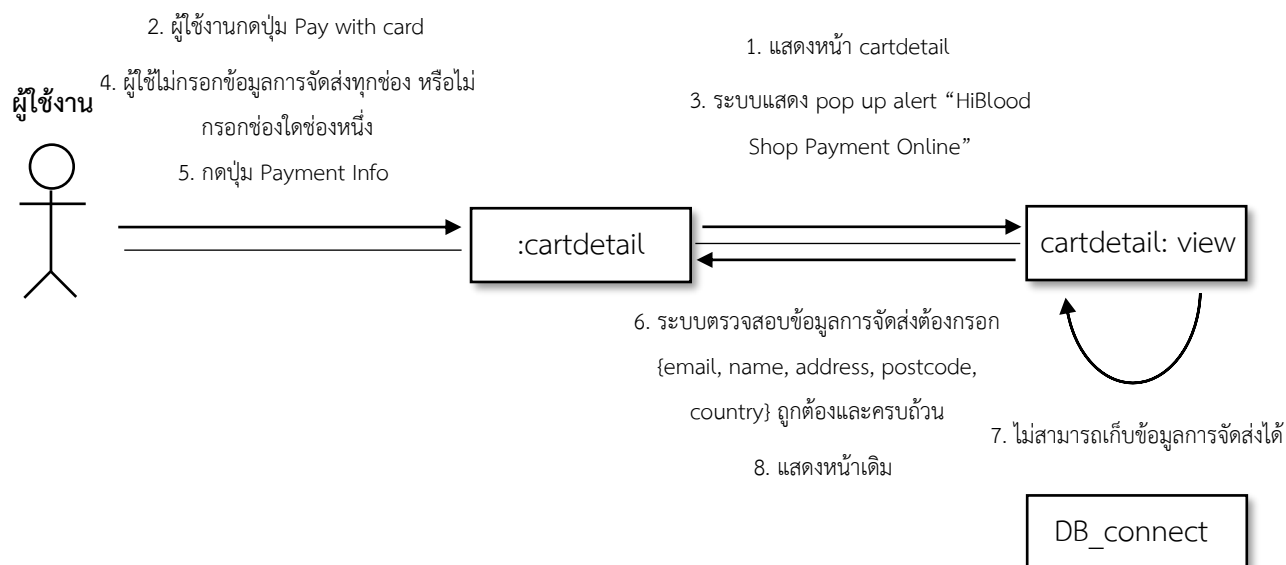
➤ Collaboration Diagram: ผู้ใช้ชำระเงิน



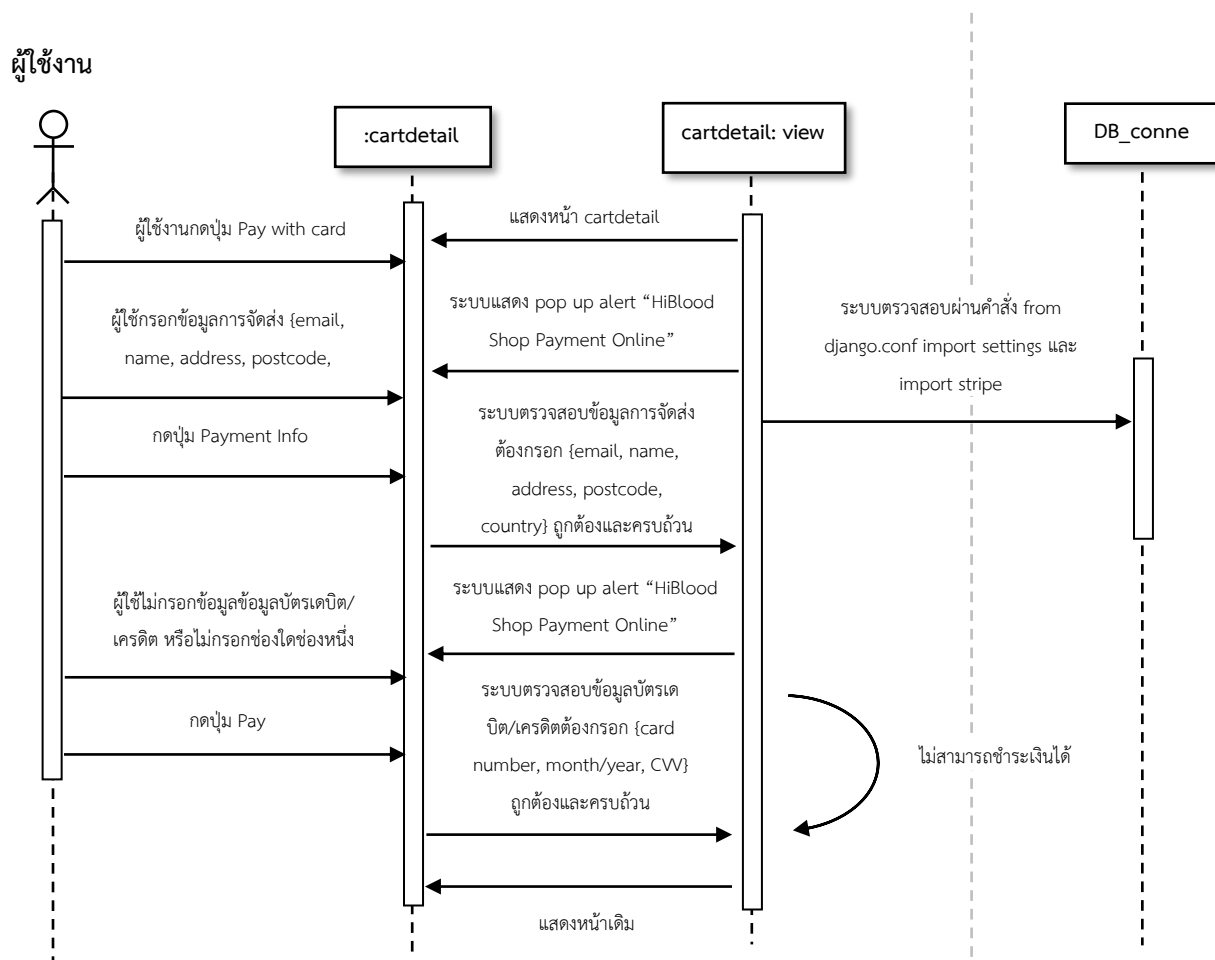
- **Sequence diagram:** ผู้ใช้ชำระเงิน กรณี ผู้ใช้ไม่กรอกข้อมูลการจัดส่งทุกช่อง หรือไม่กรอกช่องใดช่องหนึ่ง



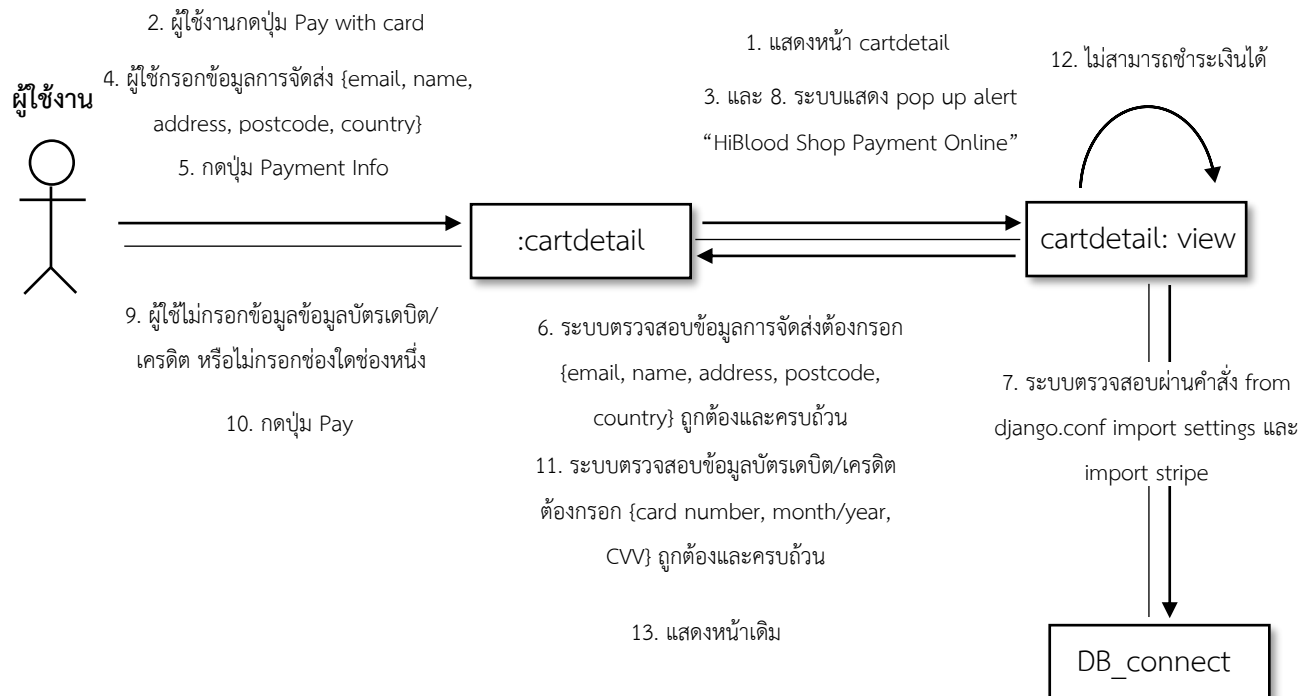
- **Collaboration Diagram:** ผู้ใช้ชำระเงิน กรณี ผู้ใช้ไม่กรอกข้อมูลการจัดส่งทุกช่อง หรือไม่กรอกช่องใดช่องหนึ่ง





- Sequence diagram: ผู้ใช้ชำระเงิน กรณี ผู้ใช้ไม่กรอกข้อมูลข้อมูลบัตรเครดิต/เครดิต หรือไม่กรอกช่องใดช่องหนึ่ง



- Collaboration Diagram: ผู้ใช้ชำระเงิน กรณี ผู้ใช้ไม่กรอกข้อมูลข้อมูลบัตรเครดิต/เครดิต หรือไม่กรอกช่องใดช่องหนึ่ง

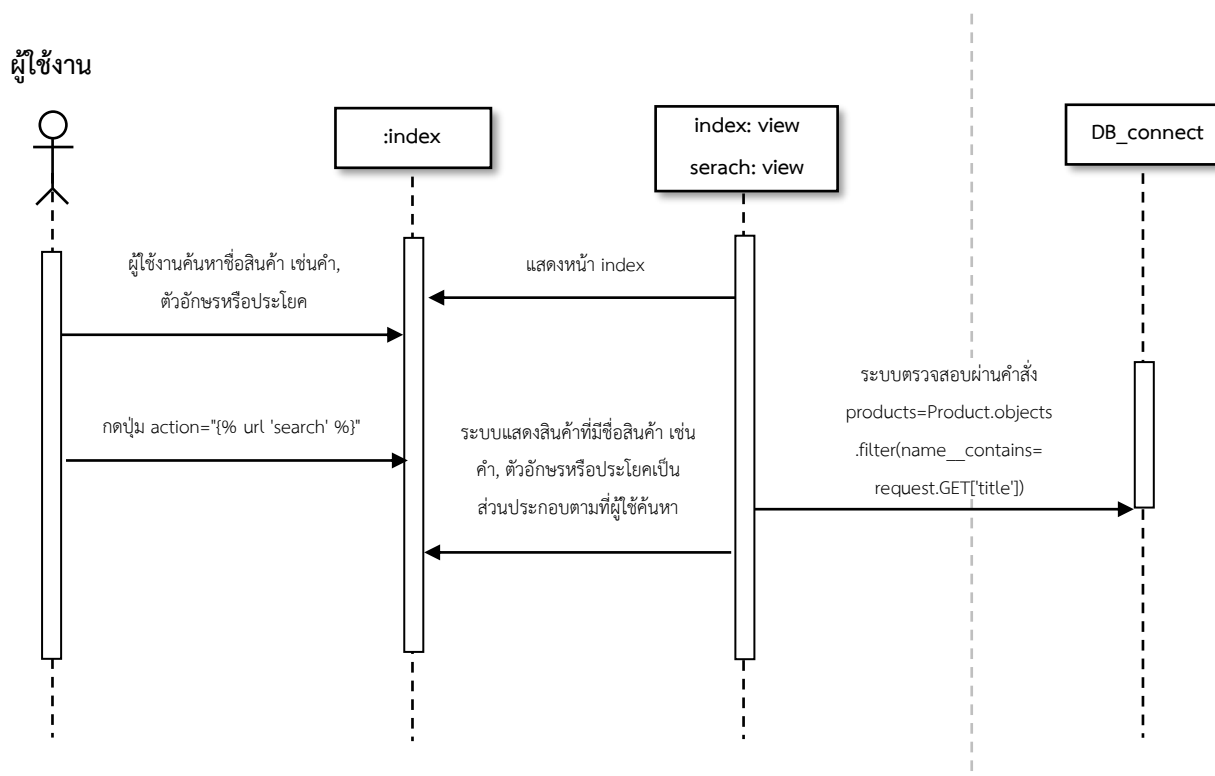


➤ Use Case Description: ผู้ใช้ค้นหาในช่อง Search

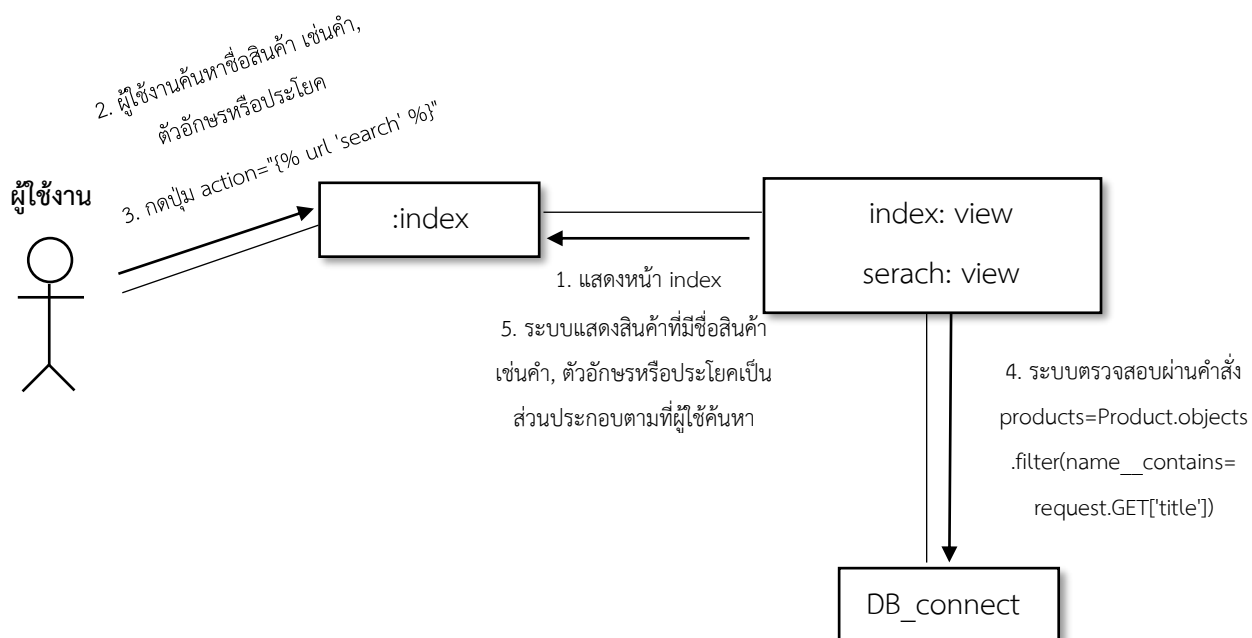
Use case	8	
Use case name	ใส่คีย์เวิร์ดในช่องค้นหา	
Actor	ผู้ใช้งาน	
Description	ผู้ใช้ทำการใส่คีย์เวิร์ดในช่องค้นหา	
Precondition	-	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้งานค้นหาซื้อสินค้า เช่น คำ, ตัวอักษรหรือประโยค  2. กดปุ่ม	3. ระบบตรวจสอบผ่านคำสั่ง <code>products=Product.objects</code> <code>.filter(name__contains=</code> <code>request.GET['title'])</code> 4. ระบบแสดงสินค้าที่มีชื่อสินค้า เช่นคำ, ตัวอักษรหรือประโยค เป็นส่วนประกอบตามที่ใช้ค้นหา
Post condition	ผู้ใช้งานค้นหาซื้อสินค้า เช่นคำ, ตัวอักษรหรือประโยคได้	
Alternative Course 1	Actor Action	System Response
	A1.1. ผู้ใช้งานค้นหาซื้อสินค้า เช่นคำ, ตัวอักษรหรือประโยคที่ไม่ได้อยู่ในร้านค้า  A1.2. กดปุ่ม	A1.3. ระบบตรวจสอบผ่านคำสั่ง <code>products=Product.objects</code> <code>.filter(name__contains=</code> <code>request.GET['title'])</code> A1.4. แสดงหน้าว่างที่ไม่มีสินค้า

ตารางที่ 3-9 Use Case Description: ผู้ใช้ค้นหาในช่อง Search

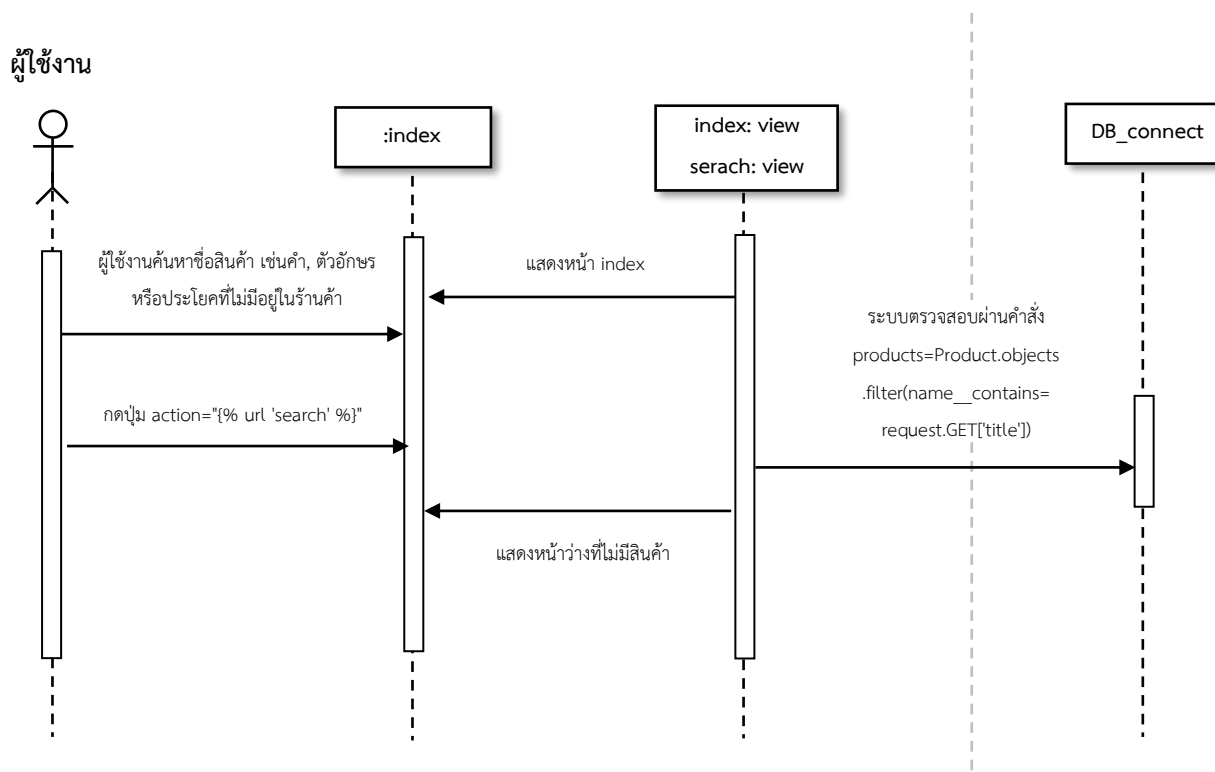
➤ Sequence diagram: ผู้ใช้ค้นหาในช่อง Search



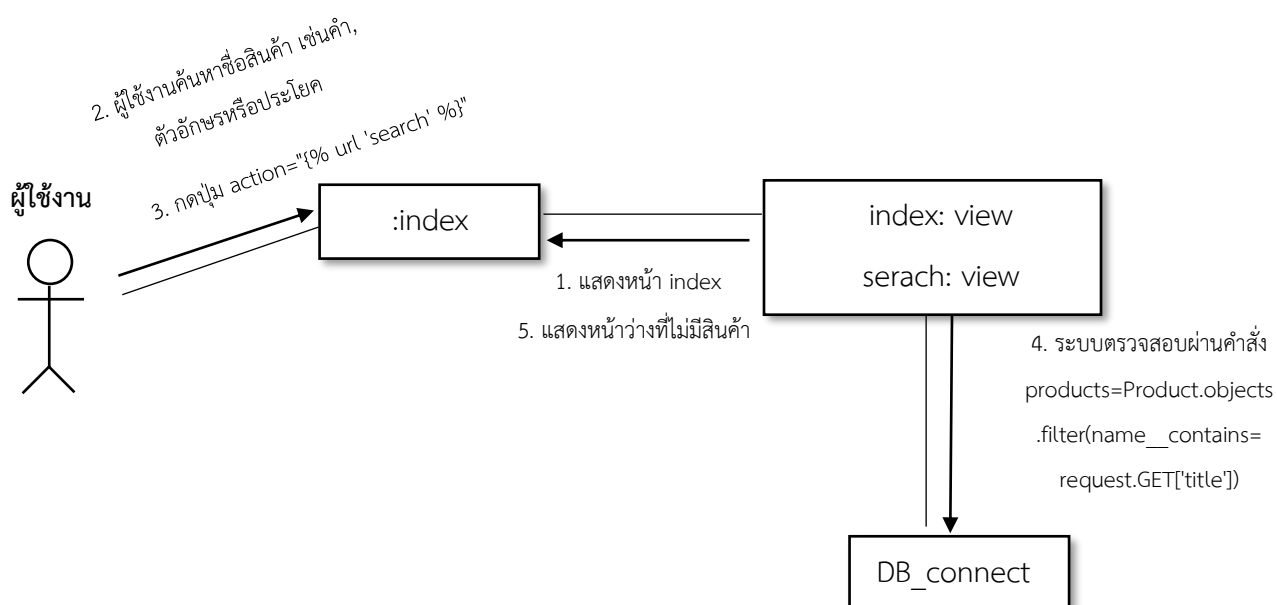
➤ Collaboration Diagram: ผู้ใช้ค้นหาในช่อง Search



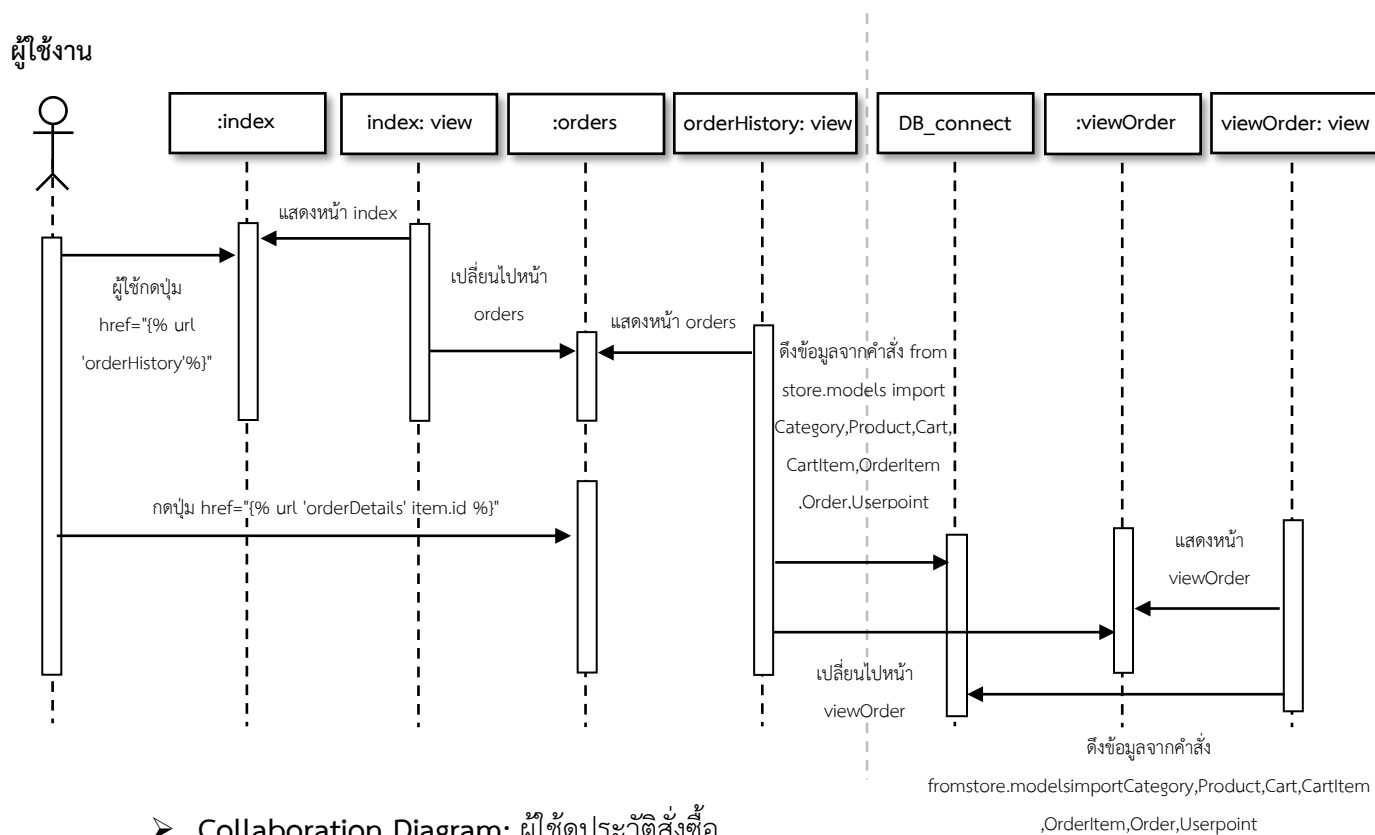
- **Sequence diagram:** ผู้ใช้ค้นหาในช่อง Search กรณี ผู้ใช้งานค้นหาชื่อสินค้า เช่นคำ, ตัวอักษรหรือประโยคที่ไม่มีอยู่ในร้านค้า



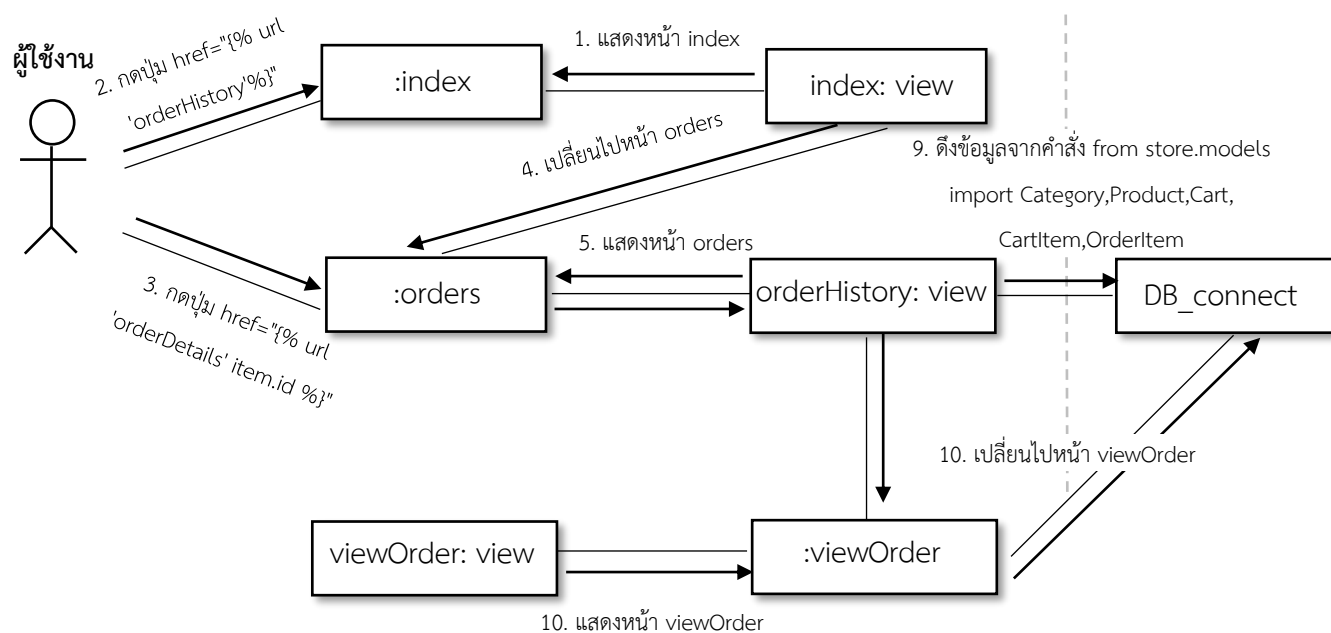
- **Collaboration Diagram:** ผู้ใช้ค้นหาในช่อง Search กรณี ผู้ใช้งานค้นหาชื่อสินค้า เช่นคำ, ตัวอักษรหรือประโยคที่ไม่มีอยู่ในร้านค้า



➤ Sequence diagram: ผู้ใช้ดูประวัติสั่งซื้อ



➤ Collaboration Diagram: ผู้ใช้ดูประวัติสั่งซื้อ



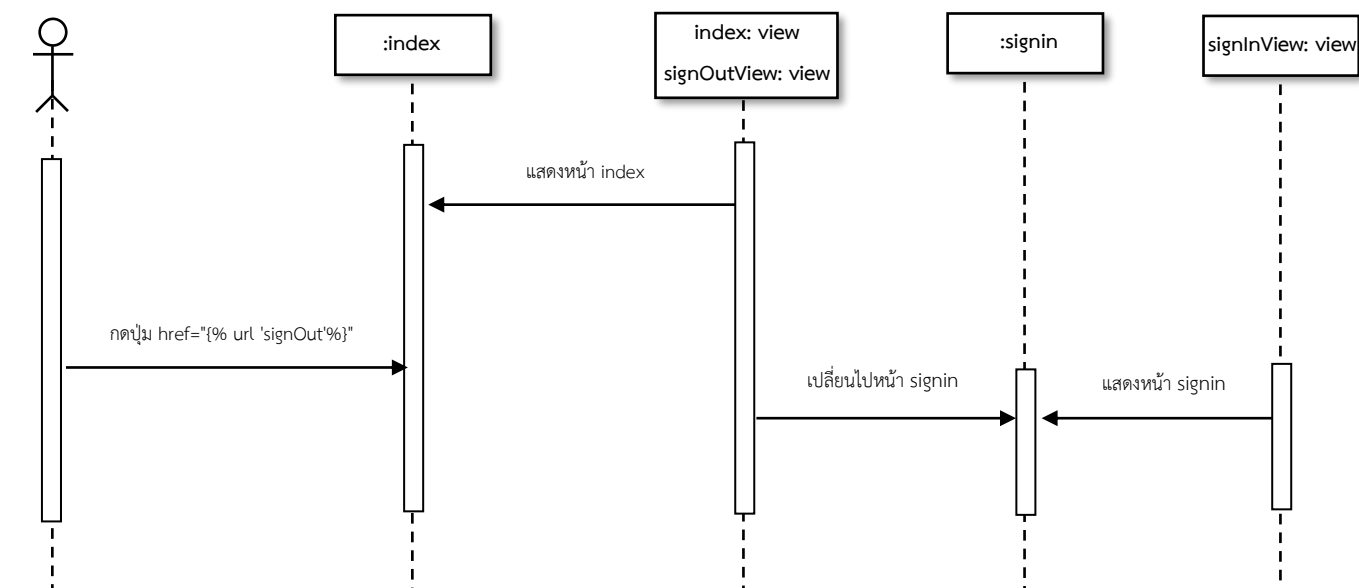
➤ Use Case Description: ผู้ใช้งาน sign out

Use case	10	
Use case name	sign out	
Actor	ผู้ใช้งาน	
Description	ผู้ใช้ทำการ sign out จากระบบ	
Precondition	ผู้ใช้ sign in	
Normal Flow	Actor Action	System Response
	1. ผู้ใช้กดปุ่ม sign out	2. เปลี่ยนหน้าไป sign in 3. แสดงหน้า sign in
Post condition	ผู้ใช้สามารถ sign out จากระบบได้และเปลี่ยนไปหน้า sign in	

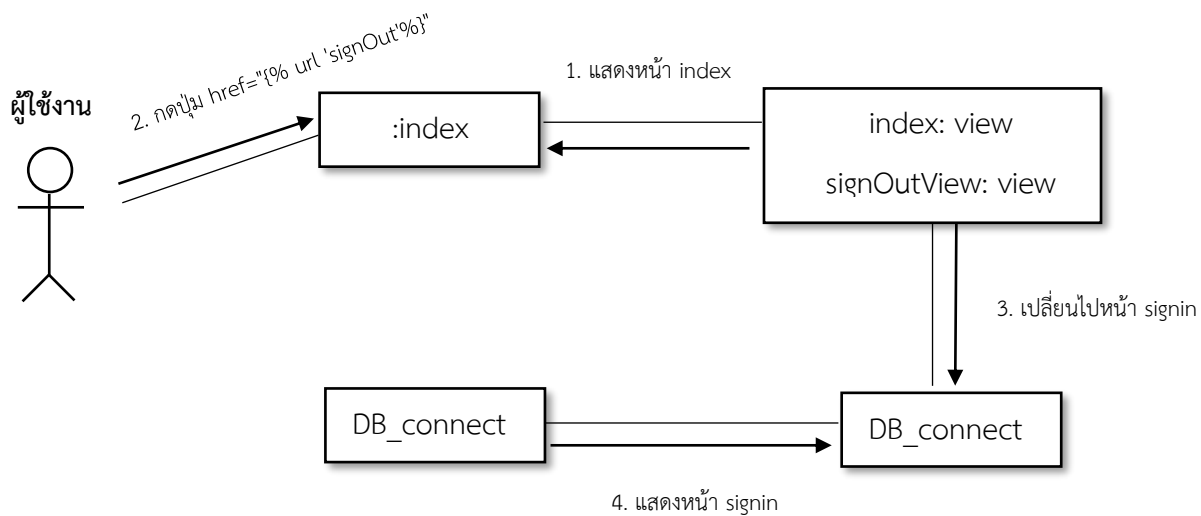
ตารางที่ 3-11 Use Case Description: ผู้ใช้งาน sign out

➤ Sequence diagram: ผู้ใช้งาน sign out

ผู้ใช้งาน



➤ Collaboration Diagram: ผู้ใช้งาน sign out

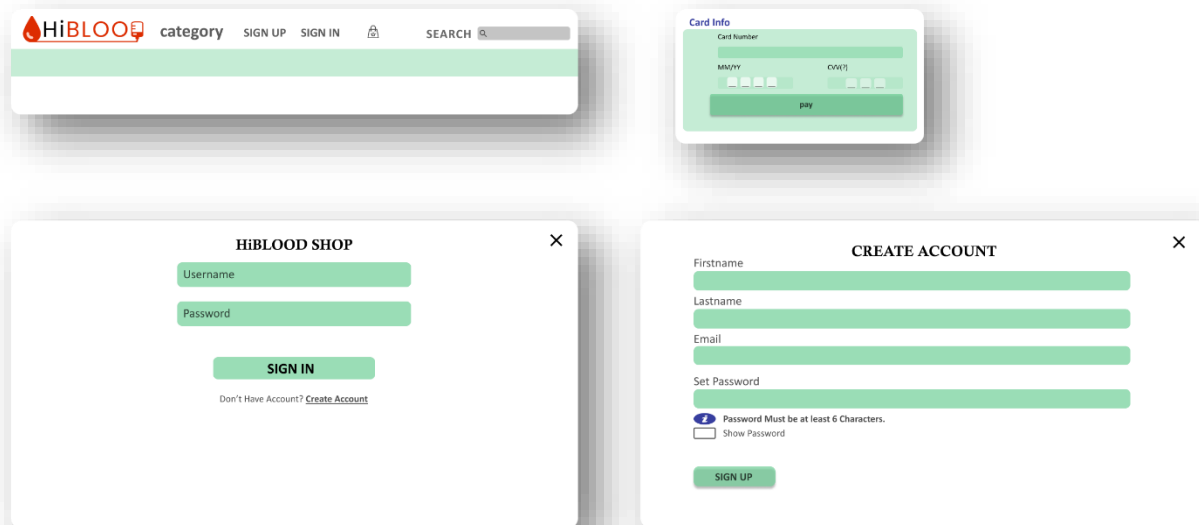


3.4. ER diagram

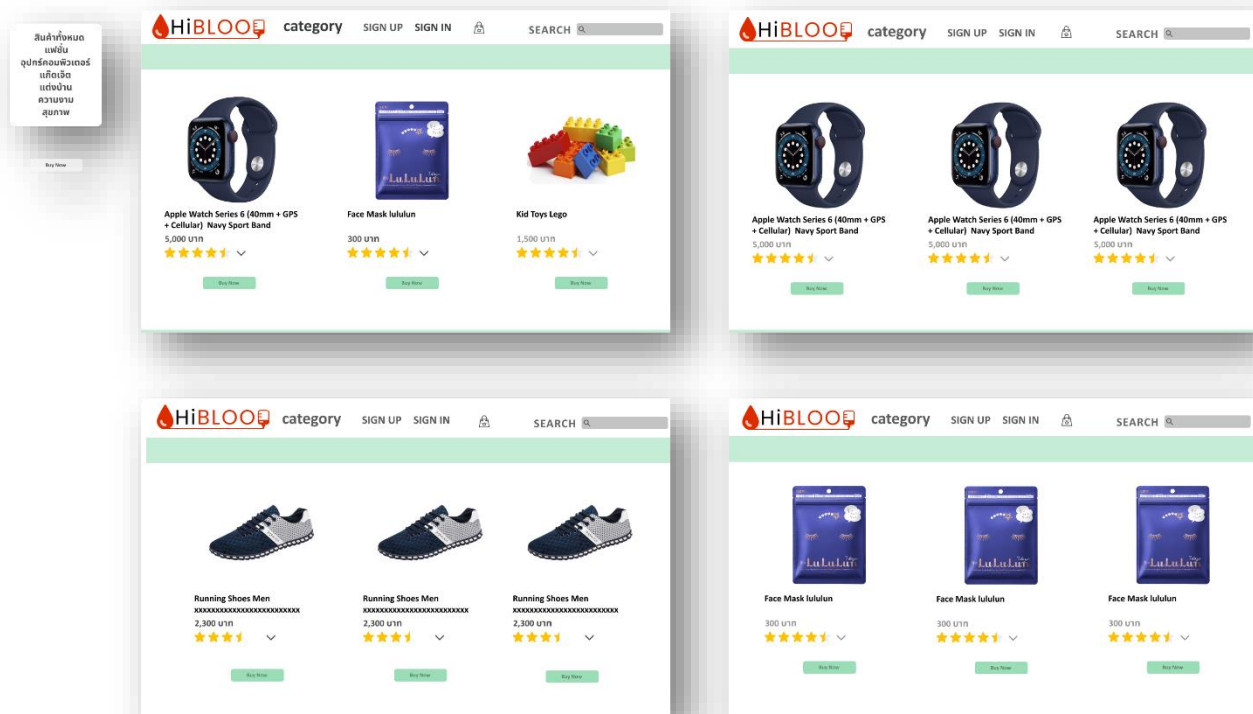


3.5. การออกแบบ User Interface

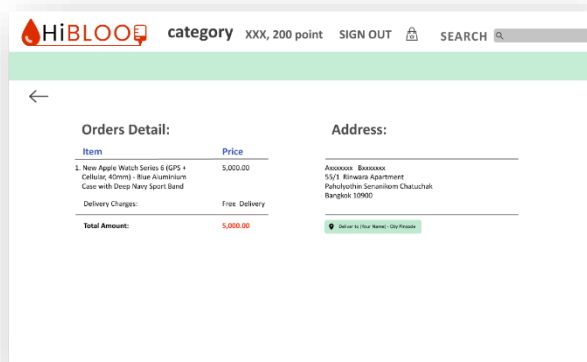
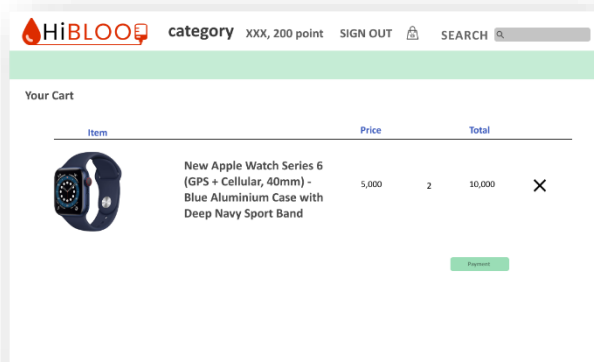
ผู้จัดทำได้ทำการออกแบบ UI โดยใช้ Figma เพื่อจำลอง web application



รูปที่ 3-3 การออกแบบส่วน navigator bar, payment, sign in และ sign up โดยใช้ Figma



รูปที่ 3-4 การออกแบบส่วน category dropdown และหมวดหมู่สินค้า โดยใช้ Figma



รูปที่ 3-5 การออกแบบส่วน cart และ order โดยใช้ Figma

บทที่ 4

การพัฒนาระบบ

จะกล่าวถึงโค้ดบางส่วนที่พัฒนา HiBLOODShop Web Application โดยเฉพาะFunction หลักๆ

4.1. Templates ของหน้าเว็บ

ในส่วนนี้เป็นการทำงานเพื่อตกแต่งและจัดหน้า User Interface ด้วยภาษา HTML ซึ่งคำสั่งรูปแบบหน้าเว็บไซต์อยู่ภายในการทำงานของแฟ้ม templates

- **layouts หลัก คือ แหล่งรวมโครงสร้างหน้าเว็บ** : ซึ่งหน้านี้เป็นแหล่งรวมโครงสร้างหน้าเว็บ ที่ได้มีการใช้ Bootstrap เข้ามาช่วยในการพัฒนาและในไฟล์นี้ได้ใช้ คำสั่ง import ไฟล์ของ navbar.html ด้วย

```
<!-- layouts หลัก คือแหล่งรวมโครงสร้างหน้าเว็บ-->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HiBLOOD Shop</title> <!-- ส่วนของการดึง BOOTSTRAP มาใช้งานแบบ cdn ,ถึงบรรทัด 10 -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" integrity="sha384-50oBUHEmpvQ+1lW4y57PTFmCaXp0ML5d60M1M7uH2+nqUivzIebhnd0JK28anvf" crossorigin="anonymous">
  <!--
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
BmbxuPwQa21c/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="
anonymous">
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.6.0/dist/umd/pop
per.min.js" integrity="sha384-
KsvD1yqQ1/1+IA7gi3P0tyJcT3vR+NdBTt13hSJ2lnve8agRGXTTyNaBYmCR/Nwi" crossorigin="
anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/js/bootstrap.min.js" integrity="sha384-
nsg8ua9HAw1y0W1btsyWgBklPnCUAFLuTMS2G72MMONqmOymq585AcH49TLBQObG" crossorigin="
anonymous"></script>
-->
</head>

<body style="background-color: #C0C0C0"> <!-- ส่วนของการปรับแต่ง GUI หน้าเว็บ, โดย
ใช้ BOOTSTRAP -->
    {% include 'navbar.html'%} <!-- ส่วนของ import file navbar.html -->

    <div class="container">
        {% block content%}

        {% endblock%}
    </div>

</body>
</html>

```

รูปที่ 4-1 โค้ด HTML โครงสร้างหน้าเว็บ

- หน้า home ที่แสดง : ไฟล์นี้เป็นการทำงานให้แสดงหน้าเว็บไซต์ที่มีรูปสินค้า, ชื่อสินค้า, คำอธิบายสินค้า, ราคาสินค้า ของสินค้าที่มีอยู่ในร้าน

```

<!-- หน้า home ที่แสดง -->
{% extends 'layouts.html' %} <!-- extend จาก layouts -->
{% load static %}
{% load humanize %}<!-- ใส่ comma ตัวเลขราคา -->

{% block content%}

<div class="container"> <!-- BOOTSTRAP -->
    <div class="row my-3">
        {% for product in products %}
            <div class="col-lg-4 col-md-6 mb-4 ">
                <div class="card h-100">

```

```

        <a href="#"></a><!-- แก้ภาพ -->
        <div class="card-body">
            <h4 class="card-title">
                <a href="{{product.get_url}}">{{product.name}}</a><!-- แก้ชื่อ -->
            </h4>
            <h5>{{product.price|intcomma}} บาท</h5> <!-- แก้ราคา -->
            <p class="card-text">{{product.description | slice:"":150 }}</p><!--
- แก้รายละเอียดสินค้า -->
        </div>
        <div class="card-footer">
            {% if product.stock <= 0 %}
                <h2 class="text-danger">สินค้าหมด</h2>

            {% else %}
                <a href="{%url 'addCart' product.id %}" class="btn btn-
primary">เพิ่มลงตะกร้า</a> <!-- href="#" # คือ ระบุการเรียกใช้งาน url -->

            {% endif %}
        </div>
    </div>
    <div>
        <div>
            {% endfor %}
        </div>
    </div>
</div>
<nav aria-label="pagination">
    {% if products.paginator.num_pages > 1 %}
        <ul class="pagination">
            {% for page_number in products.paginator.page_range %}
                <li class="page-item"
                    {% if products.number == page_number %}
                        active
                    {% endif %}
                ><a class="page-
link" href="?page={{ page_number }}"> {{ page_number }} </a></li>
            {% endfor %}
        </ul>
    {% endif %}
</nav>
</div>

{% endblock %}

```

รูปที่ 4-2 โค้ด HTML หน้า home

- navbar คือ navigator ของหน้าเว็บ : ไฟล์นี้เป็นการทำงานการจัดวางตำแหน่งของปุ่มภายในแถบนำทางได้แก่ สัญลักษณ์ของร้าน, หมวดยอดนิยมแบบย่อลงมา, ลงทะเบียน, เข้าสู่ระบบ, ตะกร้าสินค้าและช่องค้นหาสินค้า ได้มีการนำไอคอนหรือสัญลักษณ์เข้ามาตกแต่งเพื่อความสวยงามและให้เห็นได้ชัดเจนยิ่งขึ้น

```

<!-- navbar คือ navigator ของหน้าเว็บ - ย่อมาจาก layouts-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark"> <!--
- class="navbar navbar-expand-md navbar-light bg-dark ///navbar คือ menu ย่อมาจาก
โครงสร้าง layouts-->
    <!--
    <a href="/" class="navbar-brand"><i class="fas fa-heartbeat text-danger fa-
3x"></i></a>
    <button type="button" class="navbar-toggler bg-light" data-
toggle="collapse"
    data-target="#nav" name="button"><span class="navbar-toggler-icon"></span>
</button>
-->

    <div class="navbar-collapse collapse w-100 order-1 order-md-0 dual-
collapse2" id="nav"> <!-- div class="collapse navbar-collapse justify-content-
between" id="nav" /// เปลี่ยนสี ขนาด โลโก้ -->
    <ul class="navbar-nav">

        <a href="/" class="navbar-brand"><i class="fas fa-heartbeat text-
danger fa-3x"></i></a> <!-- เปลี่ยนสี ขนาด โลโก้ -->
        <button type="button" class="navbar-toggler bg-light" data-
toggle="collapse"
        data-target="#nav" name="button"><span class="navbar-toggler-
icon"></span>
</button>

        <li class="nav-item"><!-- ชื่อโลโก้ -->
            <a class="nav-link text-danger text-uppercase font-weight-bold fa-
2x">HiBlood</a>
</li>

        <li class="nav-item dropdown">
            <a class="nav-link text-light text-uppercase font-weight-bolder p-
4 dropdown-toggle " href="" data-toggle="dropdown">Category</a> <!--
- a class="nav-link text-light text-uppercase font-weight-bold px-4 dropdown-
toggle"-->
            <div class="dropdown-menu ">

```



```

        <a class="dropdown-item" href="{% url 'home' %}">สินค้าทั้งหมด</a>
    {% for category in links %}
        <a class="dropdown-
item" href="{{category.get_url}}">{{category.name}}</a>
    {% endfor %}
</div>
</li>
</ul>

<div class="navbar-collapse ">
    <ul class="navbar-nav ml-auto">
        <li class="nav-item dropdown">
            <!-- ทำการเช็ค ว่า ตอนนี้ดูค่าทำการ log in เข้ามาหรือยัง รวมถึงเช็ค ว่าได้กรอกชื่อจริงเข้ามาหรือไม่ เพราะเราจะเอาชื่อจริงมา
แสดงด้วย -->
            {% if request.user.is_authenticated and request.user.first_name %} <!--
-- Ref-part 37-38-->
            <a class="nav-link text-light text-uppercase font-weight-bold p-
4 dropdown-toggle" data-toggle="dropdown">
                Welcome, {{ request.user.first_name }} <br>
                {% if item_count >= 0 %}
                    แต้มที่เหลือ {{point}} point
                {% else %}
                    แต้มที่เหลือ 0 point
                {% endif %} <i class="fas fa-user circle"></i>
            </a>

            <div class="dropdown-menu">
                <a class="dropdown-item" href="{% url 'orderHistory'%}">ประวัติการสั่งซื้อ
</a><!-- href="" คือ การกดแล้วให้ไปที่ "ที่เรากำหนดปลายทาง" -->
            </div>
        </li>
        <li class="nav-item">
            <a class="nav-link text-light text-uppercase font-weight-bold p-
4" href="{% url 'signOut'%}"><!-- a class="nav-link text-light text-
uppercase font-weight-bold px-4" /// href="" คือ การกดแล้วให้ไปที่ "ที่เรากำหนดปลายทาง" -->
                Sign Out <i class="fas fa-sign-out-alt"></i> <!--
sign out / # from views.py(2B), now navbar.html(3B) / part39 -->
            </a>
        </li>
        <!-- ทำการเช็ค ว่า กรณีไม่มีการ log in ก็จะแสดง SignUp กับ SignIn -->
        {% else %}
        <li class="nav-item">
            <a class="nav-link text-light text-uppercase font-weight-bold p-
4" href="{% url 'signUp'%}"><!-- a class="nav-link text-light text-
uppercase font-weight-bold px-4" /// href="" คือ การกดแล้วให้ไปที่ "ที่เรากำหนดปลายทาง" -->

```

```

        Sign Up <i class="fas fa-user-plus"></i> <!--sign up-->
    </a>
</li>

    <li class="nav-item">
        <a class="nav-link text-light text-uppercase font-weight-bold p-
4" href="{% url 'signIn'%}"><!-- a class="nav-link text-light text-
uppercase font-weight-bold px-4"//เดิมจาก signin.html(3A)-นี่คือ 4A , href="" คือ การกดแล้ว
ให้ไปที่ "ที่เรากำหนดปลายทาง(ในที่นี้คือ อ้างอิงผ่านชื่อ(ที่กำหนดชื่อ path ใน urls.py) เพื่อจะได้ไม่เกิด path ที่อาจซ้อนกัน)" -->
        Sign In <i class="fas fa-sign-in-alt"></i> <!--sign in-->
    </a>
</li>
{% endif %}
</ul>
</div>

<!-- ตะกร้าสินค้า -->
<div class="navbar-collapse ">
    <ul class="navbar-nav ml-auto">
        <li class="nav-item">
            <a class="nav-link text-light text-uppercase font-weight-bold p-
4" href="{% url 'cartdetail'%}"><!-- href="" คือ การกดแล้วให้ไปที่ "ที่เรากำหนดปลายทาง" -->
                <i class="fas fa-shopping-cart"></i>
                {% if item_count > 0%}
                    {{item_count}}
                {% else %}
                    0
                {% endif %}
            </a>
        </li>
    </ul>
</div>
<!-- ช่อง Search เพื่อค้นหา-->
<form class="form-inline ml-
3 " action="{% url 'search' %}", method="GET"> <!--
- set url ใน action / from views.py(2C), now navbar.html(3C)-->
    <div class="input-group">
        <input type="text" name="title" class="form-
control" placeholder="Search">
        <div class="input-group-append">
            <button type="submit" class="btn btn-danger"><i class="fas fa-
search"></i></button>
        </div>
    </div>
</form>

```

```
</div>
</nav>
```

รูปที่ 4-3 โค้ด HTML ส่วน navigator bar

4.2. การจัดส่วนของ Django Administrator หรือ Admin Dashboard

ในส่วนนี้ทาง Django จะจัดการทำงานของหน้า Administrator มาให้กับ Admin เพื่อให้ทางผู้พัฒนาไม่จำเป็นต้องสร้างคำสั่งส่วนขึ้นมาเองให้ลำบาก แต่ผู้พัฒนายังสามารถจัดหรือตกแต่งหน้า Django Administrator ให้แสดงผลออกมาตามที่ต้องการได้ด้วย เช่น กำหนดการแสดงผลรายการสินค้าในหน้า Admin Dashboard มา 5 รายการต่อหน้า ซึ่งคำสั่งอยู่ภายใต้การทำงานของไฟล์ admin.py

```
# class การทำงานของ Admin Dashboard
from django.contrib import admin # ลงทะเบียนหรืออัพโหลดขึ้น db และ แสดงส่วน admin dashboard ด้วย
from store.models import Category,Product, Cart, CartItem, Order, OrderItem, Userpoint

# Register your models here.

# ปรับแต่ง Product Admin / ทำการแสดงผลของสินค้า โดยไม่ต้องคลิกเข้าไปสินค้านั้นก็ได้
class ProductAdmin(admin.ModelAdmin):
    # ดูได้จาก ฐานข้อมูลใน sql
    list_display=['name', 'price', 'stock', 'created', 'updated']
    # ทำให้สามารถแก้ไขข้อมูล ได้ที่หน้านั้นเลย โดยไม่ต้องกดเข้าไปในสินค้า
    list_editable=['price', 'stock']
    # ดูได้จาก ฐานข้อมูลใน sql
    list_per_page=5

class OrderAdmin(admin.ModelAdmin):
    # ดูได้จาก ฐานข้อมูลใน sql
    list_display=['id', 'name', 'email', 'total', 'token', 'created', 'updated']
    list_per_page=5

class OrderItemAdmin(admin.ModelAdmin):
    # ดูได้จาก ฐานข้อมูลใน sql
    list_display=['order', 'product', 'quantity', 'price', 'created', 'updated']
    list_per_page=5

class UserPointsAdmin(admin.ModelAdmin):
    # ดูได้จาก ฐานข้อมูลใน sql
    list_display=['name', 'totalp']
    list_per_page=5

admin.site.register(Category)
admin.site.register(Product, ProductAdmin)
admin.site.register(Cart)
```

```
admin.site.register(CartItem)
admin.site.register(Order, OrderAdmin)
admin.site.register(OrderItem, OrderItemAdmin)
admin.site.register(Userpoint, UserPointsAdmin)
```

รูปที่ 4-4 โค้ดการทำงานของ Admin Dashboard

4.3. ส่วนของการจำกัดความการแสดงผลของหน้าเว็บนั้นๆ

ในการเรียกให้แสดงผลหน้าเว็บ ผู้จัดทำมีการเขียนคำสั่งการจำกัดความว่าให้หน้าเว็บที่ถูกเรียกขึ้นมาขึ้นนั้นแสดงผลออกมาอย่างไร ที่จะตรงกับการเรียกใช้ของผู้ใช้ระบบเพื่อไม่ให้เกิดความสับสนหรือเกิดความผิดพลาด ซึ่งคำสั่งการจำกัดความอยู่ภายใต้การทำงานของไฟล์ views.py

- การแสดงหน้าแรก

ในส่วนของหน้าแรกเราจะแสดงสินค้าที่อยู่ในร้านค้า ผู้จัดทำได้กำหนดการแสดงผลสินค้า 6 ชิ้นต่อ 1 หน้า มีการแสดงสินค้าตามหมวดหมู่ที่เลือก แสดงสินค้าตามเลขหน้าที่จัดการ

```
def index(request, category_slug=None): # หน้าแรก
    products=None
    category_page=None
    # ถ้าค่า category_slug != None
    if category_slug != None:
        # ทำการค้นหาหมวดหมู่ category
        category_page=get_object_or_404(Category,slug=category_slug)
        # ดึงสินค้าพร้อมกับหมวดหมู่
        products=Product.objects.all().filter(category=category_page,available=
True)
    else :
        products=Product.objects.all().filter(available=True)

    # set ว่าจะแบ่งสินค้า จำนวนกี่ชิ้น/หน้า
    # ex 8 / 3 = 3 page
    paginator=Paginator(products,6)
    # set เลขหน้าเป็นค่าเริ่มต้น
    try:
        page=int(request.GET.get('page','1'))
    except:
        page=1
    # ค่าเริ่มต้นที่กำหนดมา set ให้กับ paginator
    try:
        productperPage=paginator.page(page)
    except(EmptyPage,InvalidPage):
```

```

productPerPage=paginator.page(paginator.num_pages)

return render(request, 'index.html', {'products':productPerPage, 'category':category_page})
#Hello Hiblood Donation

```

รูปที่ 4-5 โค้ดการจำกัดความการทำงานหน้าแรก

- การจำกัดความของการแสดงหน้าตะกร้าสินค้า

ผู้จัดทำได้จำกัดความในหน้าตะกร้าสินค้า หากมีการเรียกใช้หน้าตะกร้าสินค้าโดยเพิ่มสินค้าลงตะกร้าแต่ยังไม่มีกรเข้าสู่ระบบ ระบบจะแสดงหน้าให้เข้าสู่ระบบก่อนการเพิ่มสินค้าลงตะกร้า และส่วนของการอัปเดตจำนวนสินค้าในตะกร้าตามที่ผู้ใช้ได้เพิ่มสินค้าเข้ามาให้แสดงบนแถบนำทางด้วย

```

# หากมีการ เรียกการใช้งาน def add cart แล้วยังไม่มีการ เข้าใช้งานระบบ(log in) จะต้องมีการ redirect ไป
หน้า log in
@login_required(login_url='signIn') # มีการระบุ path ของ log in
def addCart(request,product_id):
    # รหัสสินค้า
    # ดึงสินค้าตามรหัสที่ส่งมา
    product=Product.objects.get(id=product_id)

    # สร้างตะกร้าสินค้า
    try:
        cart=Cart.objects.get(cart_id=_cart_id(request))
    # เมื่อไม่พบ object cart
    except Cart.DoesNotExist:
        cart=Cart.objects.create(cart_id=_cart_id(request))
        # บันทึกลงฐานข้อมูล
        cart.save()

    # เมื่อมีตะกร้าสินค้าแล้ว
    try:
        # ชื่อรายการสินค้าซ้ำ
        cart_item=CartItem.objects.get(product=product,card=cart)
        if cart_item.quantity < cart_item.product.stock:
            # เปลี่ยนจำนวน quantity
            cart_item.quantity += 1
            # บันทึก / อัปเดตค่า
            cart_item.save()
        except CartItem.DoesNotExist:
            # ชื่อรายการสินค้าครั้งแรก
            # บันทึกลงฐานข้อมูล

```

```

    cart_item=CartItem.objects.create(
        product=product,
        cart=cart,
        quantity=1
    )
    cart_item.save()
    return redirect('cartdetail')

```

รูปที่ 4-6 โค้ดจำกัดความการทำงาน หน้าตะกร้าสินค้า

- การจำกัดความของการแสดงหน้าแบบฟอร์มลงทะเบียน

เป็นส่วนของหน้าการลงทะเบียนสร้างบัญชีผู้ใช้ ผู้จัดทำได้มีการ import คำสั่งการใช้แบบฟอร์มของ Django เข้ามาช่วยและจะปรากฏช่องให้ผู้กรอกข้อมูลโดยมีชื่อจริง, นามสกุล, ชื่อผู้ใช้, อีเมล, รหัสผ่านและการยืนยันรหัสผ่านอีกครั้ง ซึ่งคำสั่งการจำกัดความนี้จะเช็คความถูกต้องของแบบฟอร์มที่ผู้ใช้ทำการบันทึกเข้ามา เมื่อเรียบร้อยแล้วแสดงไปหน้าเข้าสู่ระบบ

```

def signUpView(request): # กรณียังไม่มีการล็อกอิน / ลงทะเบียน
    #อ้างอิงไปยัง field ที่กำหนด
    if request.method=='POST':
        form=SignUpForm(request.POST)
        #เช็คความถูกต้องของแบบฟอร์ม
        if form.is_valid():
            #บันทึกข้อมูล User ที่โยนมาจากหน้าเว็บ
            form.save()
            #บันทึก Group Customer
            #ดึง username จากแบบฟอร์มมาใช้
            username=form.cleaned_data.get('username')
            #ดึงข้อมูล user จากฐานข้อมูล
            signUpUser=User.objects.get(username=username)
            #จัด Group โดยไปดึง Group มาทำงานก่อน
            customer_group=Group.objects.get(name="Customer")
            #นำ Group ที่ดึงมาจากฐานข้อมูล กำหนดให้กับ user
            customer_group.user_set.add(signUpUser)
            return redirect('signIn') #FIX
        else :
            form=SignUpForm()
            return render(request,"signup.html",{ 'form':form}) #FIX /// return render(r
equest,"signIn.html",{ 'form':form})

```

รูปที่ 4-7 โค้ดจำกัดความการทำงานของหน้าแบบฟอร์มลงทะเบียน

- การจำกัดความการลบสินค้าในหน้าตะกร้าสินค้า

หากผู้ใช้ไม่ต้องการสินค้าในตะกร้าที่เพิ่มเข้ามาแล้วต้องการจะนำออกจากตะกร้า คำสั่งจำกัดความนี้จะทำงานโดยเข้าถึงสินค้าภายในตะกร้าแล้วค้นหาสินค้าหากตรงกันจึงทำการลบ เมื่อลบแล้วแสดงหน้าตะกร้าสินค้าและรายการสินค้าปัจจุบัน

```
def removeCart(request,product_id): # ลบของในตะกร้าสินค้า
    # ทำงานกับตะกร้าสินค้า
    cart=Cart.objects.get(cart_id=_cart_id(request)) # เข้าถึงตะกร้าสินค้าได้แล้ว
    # ทำงานกับสินค้าที่จะลบ ex product id = 1
    product=get_object_or_404(Product,id=product_id) # เข้าถึงสินค้าได้แล้ว ด้วย product_id
    # เช็คว่ารหัสสินค้าใดเป็น 1 แล้วอยู่ในตะกร้าสินค้าที่กำหนด
    cartItem=CartItem.objects.get(product=product,card=cart)
    # ลบรายการสินค้า 1 ออกจากตะกร้า A โดยลบจาก รายการสินค้าในตะกร้า (CartItem)
    cartItem.delete()
    # เมื่อลบเสร็จก็สั่งอยู่ที่หน้าตะกร้าสินค้าเดิม
    return redirect('cartdetail')
```

รูปที่ 4-8 โค้ดจำกัดความการลบสินค้าในหน้าตะกร้าสินค้า

4.4. การกำหนดเส้นทาง

ส่วนนี้ทาง Django จัดการมาให้เรา เพียงแต่เราต้องเพิ่มการคำสั่งการกำหนดเส้นทางของหน้าเว็บไซต์

```
# Django จัดมาให้ // ส่วนการสร้าง path
"""django workshop URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""

from django.contrib import admin
from django.urls import path
from store import views #import จาก view ใน store
```

```

from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name="home"),
    path('category/<slug:category_slug>', views.index, name="product_by_category"
),
    path('product/<slug:category_slug>/<slug:product_slug>', views.productPage, n
ame='productDetail'),
    path('cart/add/<int:product_id>', views.addCart, name="addCart"),
    path('cartdetail/', views.cartdetail, name="cartdetail"),
    path('cart/remove/<int:product_id>', views.removeCart, name="removeCart"),
    path('account/create', views.signUpView, name="signUp"),
    path('account/login', views.signInView, name="signIn"), # name="signIn" = อ้างอิง
ไปยัง path account login / start 1A ไปต่อที่ views.py(2A)
    path('account/logout', views.signOutView, name="signOut"), # name="signout" =
อ้างอิงไปยัง path log out / start 1B ไปต่อที่ views.py(2B) / part39
    path('serach/', views.serach, name='search'), # name="search" = อ้างอิงไป
ยัง path serach / start 1C ไปต่อที่ views.py(2C)
    path('orderHistory/', views.orderHistory, name='orderHistory'),
    path('order/<int:order_id>', views.viewOrder, name='orderDetails'),
    path('cart/thankyou', views.thankyou, name='thankyou')
]

#product/fashion/shoes
# เช็คว่าเราเปิด DEBUG หรือไม่
if settings.DEBUG :
    # /media/product/
    urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
    # /static/
    urlpatterns+=static(settings.STATIC_URL,document_root=settings.STATIC_ROOT)
    # /static/media/product/bear.jpg

```

รูปที่ 4-9 โค้ดการกำหนดเส้นทาง

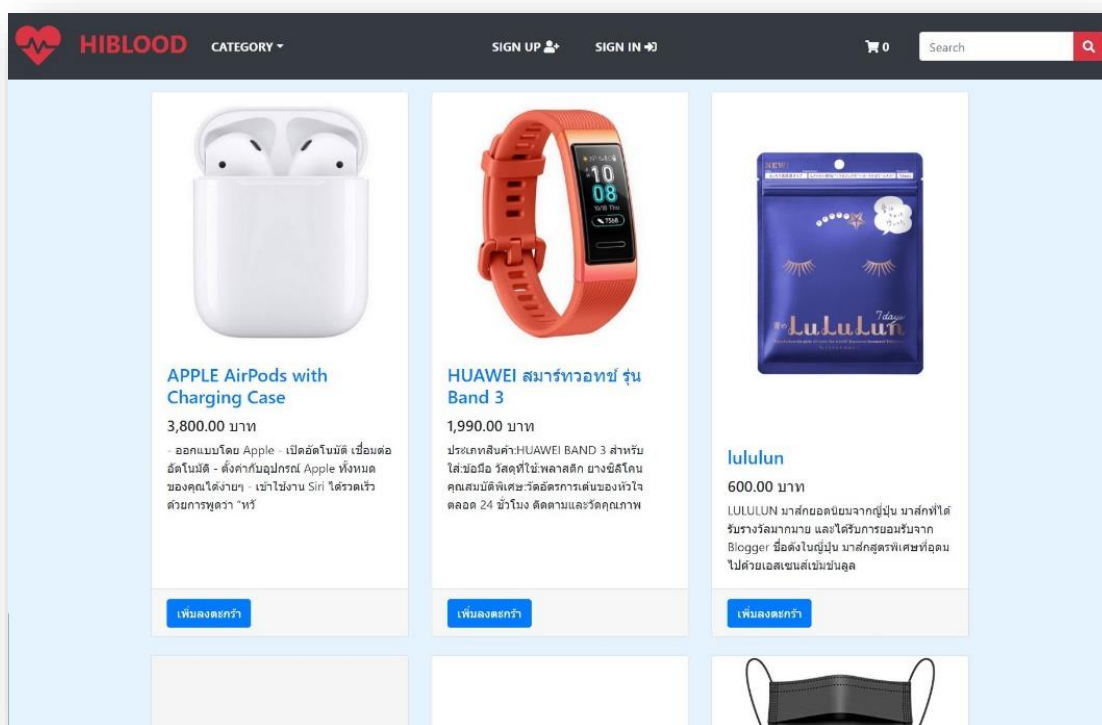
บทที่ 5

ผลการทดสอบระบบ

5.1 . ผลการทดลองใช้โปรแกรม

- Front-end ของผู้ใช้

หน้าแรกเว็บจะแสดงสินค้าภายในร้านจำนวน 6 ชิ้นต่อหน้า มีเลขหน้าเพื่อดูรายการสินค้าอื่น ๆ แต่ไม่สามารถเพิ่มสินค้าลงตะกร้าได้



รูปที่ 5-1 หน้าแรกเมื่อผู้ใช้เข้าเว็บไซต์

เมื่อผู้ใช้กดปุ่มเข้าสู่ระบบหรือ sign in บนแถบนำทางก็จะเปลี่ยนไปหน้าลงชื่อเข้าใช้ระบบ HiBLOOD SHOP ให้ผู้ใช้กรอก Username และ Password

รูปที่ 5-2 หน้า sign in

เมื่อผู้ใช้กดปุ่มลงทะเบียนหรือ sign up บนแถบเมนูนำทาง ก็จะเปลี่ยนไปหน้าแบบฟอร์มลงทะเบียนให้ผู้ใช้กรอกครบทุกช่องและกรอกให้ตรงตามที่กำหนด ถ้ากรอกไม่ถูกต้องตามที่กำหนดจะแสดงแจ้งเตือน

รูปที่ 5-3 หน้า sign up

HIBLOOD CATEGORY **SIGN UP** **SIGN IN** 0 Search

แบบฟอร์มลงทะเบียน

First name*
suchada

Last name*
malaiwet

Username*
pjoy ⓘ
A user with that username already exists.
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*
pjoy@gmail ⓘ
Enter a valid email address.
example@gmail.com

Password*

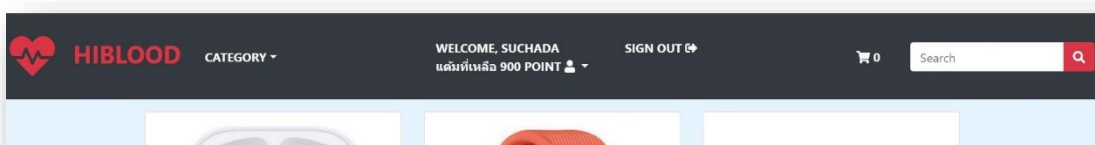
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*
 ⓘ
 This password is too short. It must contain at least 8 characters.
 This password is too common.
 Enter the same password as before, for verification.

Register

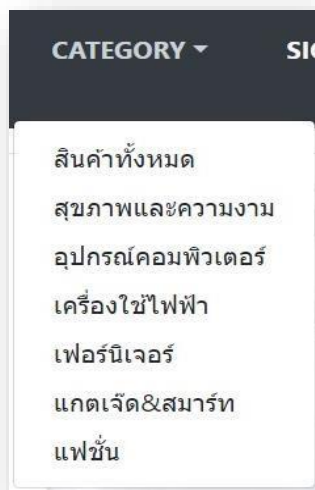
รูปที่ 5-4 แจ้งเตือนเมื่อผู้ใช้กรอก Username ที่ถูกใช้ลงทะเบียนไปแล้ว, กรอก email ผิดรูปแบบ, กรอก Password ไม่ตรงตามเกณฑ์การตั้งและน้อยกว่า 8 ตัวอักษร

เมื่อผู้ใช้ทำการ sign in หรือ log in เข้าเว็บไซต์ ตรง sign in ก็จะเปลี่ยนเป็นชื่อผู้ใช้ใน navigator bar จะแสดงหมวดหมู่สินค้า, ชื่อผู้ใช้, แต้มที่เหลือ, ตะกร้าสินค้า, sign out เพื่อกำหนดระบบและช่องค้นหาสินค้า



รูปที่ 5-5 หน้าแรกเมื่อผู้ใช้ sign in เข้ามา

เมื่อผู้ใช้กดปุ่มหมวดหมู่หรือ category จะแสดงหมวดหมู่สินค้าและเมื่อกดที่ชื่อผู้ใช้จะแสดงประวัติการสั่งซื้อ



รูปที่ 5-6 category แสดงย่อยหมวดหมู่



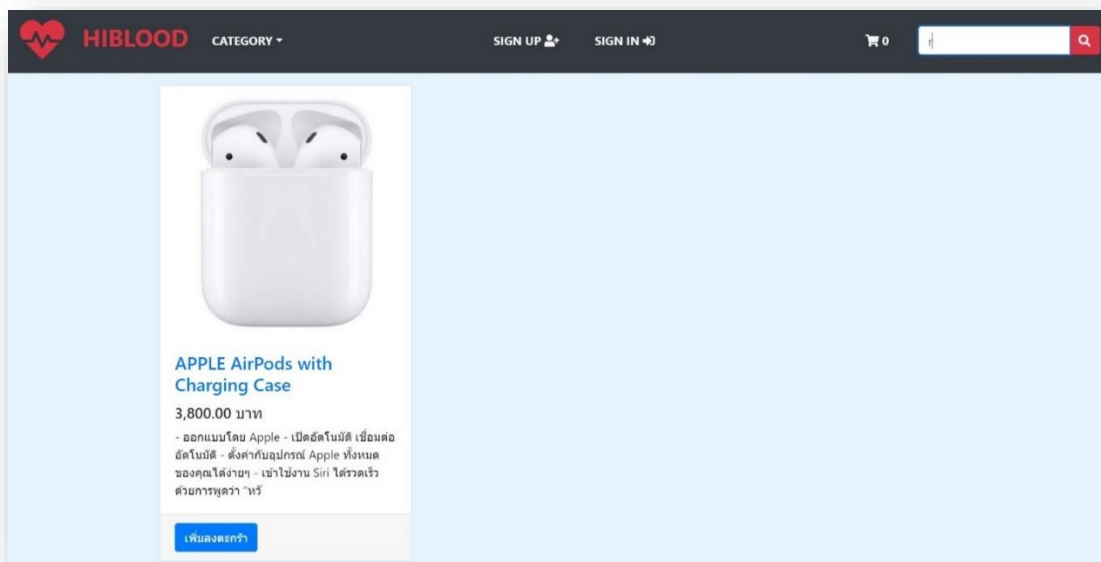
รูปที่ 5-7 ชื่อผู้ใช้แสดงย่อยประวัติการสั่งซื้อ

เมื่อผู้ใช้กดปุ่มตะกร้าสินค้าหรือสัญลักษณ์รูปตะกร้า หากผู้ใช้ยังไม่ได้มีการเพิ่มสินค้าลงตะกร้า จะแสดงแจ้งเตือนว่า “ไม่มีสินค้าในตะกร้า”



รูปที่ 5-8 แจ้งเตือนของตะกร้าสินค้าเมื่อผู้ใช้ยังไม่มีสินค้าในตะกร้า

ในส่วนของการค้นหาสามารถใส่ได้ทั้งคำ, ประโยค, ตัวอักษรแล้วก็จะแสดงสินค้านั้นๆ



รูปที่ 5-9 แสดงผลเมื่อผู้ใช้งานค้นหาตัวอักษร “r”

การแสดงผลเมื่อผู้ใช้งานเข้าไปในสินค้าที่สนใจก็จะแสดงรายละเอียดข้อมูลสินค้าแบบเต็ม



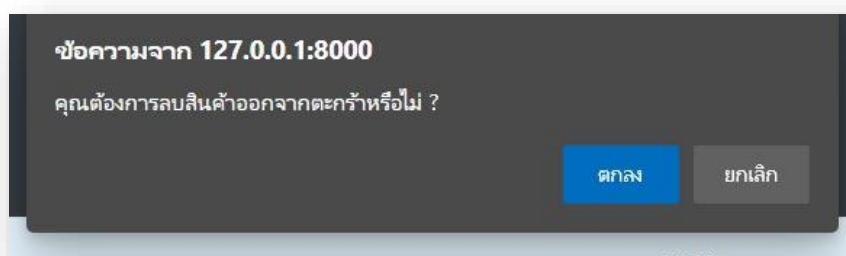
รูปที่ 5-10 แสดงผลเมื่อผู้คลิกสินค้า Apple airpods with charging case

เมื่อผู้ใช้เพิ่มสินค้าลงตะกร้า หน้าตะกร้าสินค้าขึ้นแสดงรายการสินค้าตามจำนวนที่ผู้ใช้เพิ่มเข้ามา

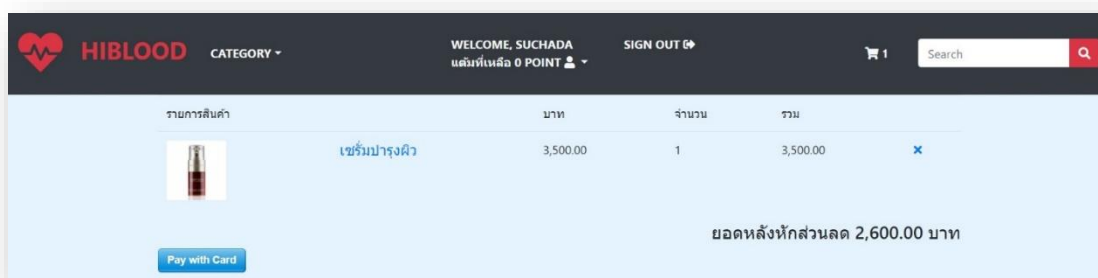


รูปที่ 5-11 ผู้ใช้เพิ่มสินค้าลงตะกร้า 2 รายการ

หากผู้ใช้ลบสินค้ารายการนั้นจะมีแจ้งเตือน “คุณต้องการลบสินค้าออกจากตะกร้าหรือไม่” ถ้าผู้ใช้กดตกลงสินค้าก็จะถูกลบออกจากตะกร้า

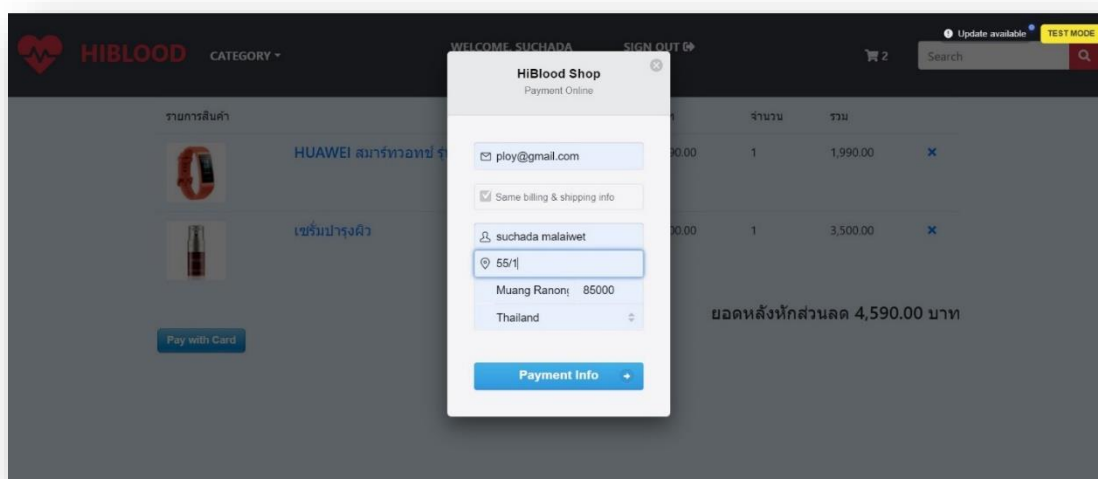


รูปที่ 5-12 แจ้งเตือนเมื่อผู้ใช้ต้องการลบสินค้าออกจากตะกร้า

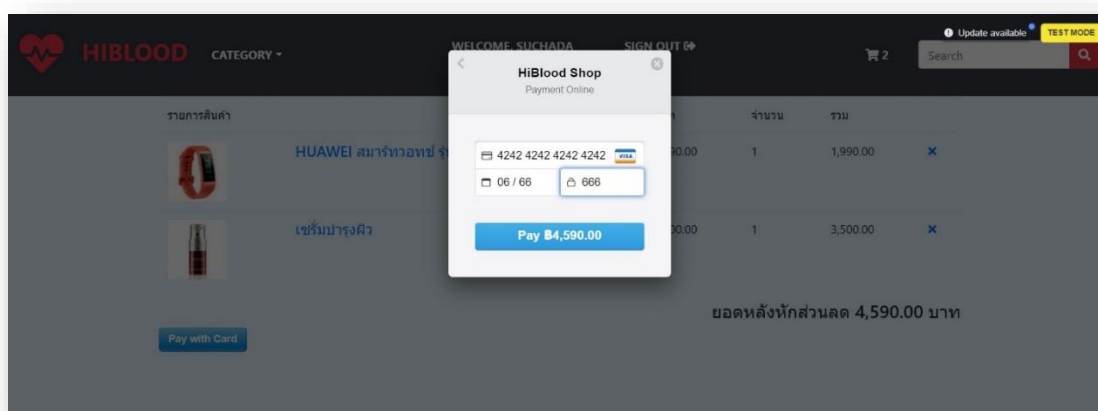


รูปที่ 5-13 สินค้าถูกลบออกจากตะกร้าเรียบร้อยแล้ว

เมื่อผู้ใช้พอใจกับสินค้าในตะกร้าแล้วก็กดปุ่ม Pay with card ในยอดนี้หากผู้ใช้แต่้มคะแนนไม่ถึงตามราคาสินค้านั้น ก็จะมีส่วนต่างที่ต้องชำระ โดยทางเว็บจะมีการชำระแบบผ่านบัตรเท่านั้น ซึ่งจะมีหน้าต่างแสดงขึ้นมาให้กรอกข้อมูลชื่อผู้ใช้, ที่อยู่และเมื่อกด Payment Info จะให้กรอกระายละเอียดของบัตรที่จะทำการชำระเงิน

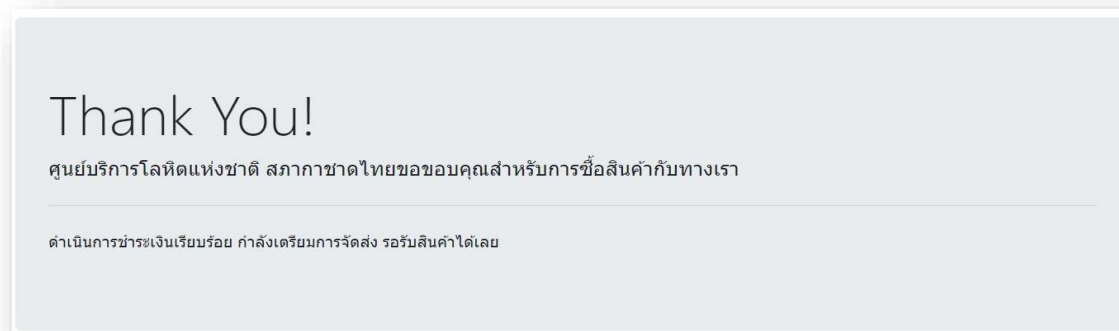


รูปที่ 5-14 ผู้ใช้กรอกรายละเอียดการจัดส่ง



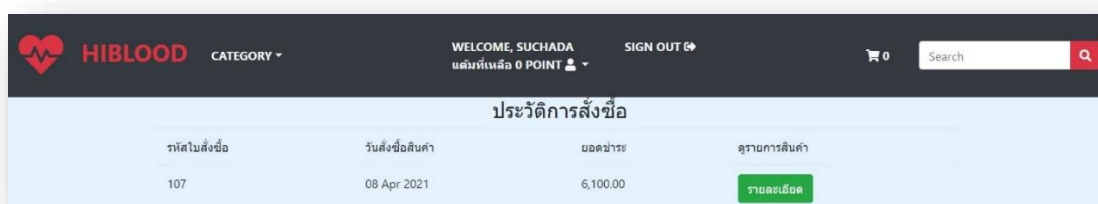
รูปที่ 5-15 ผู้ใช้กรอกรายละเอียดบัตรเพื่อชำระเงิน

เมื่อการสั่งซื้อสินค้าสำเร็จ จะแสดงผลไปยังหน้า Thank You! และรายละเอียดภายในหน้า Thank You!

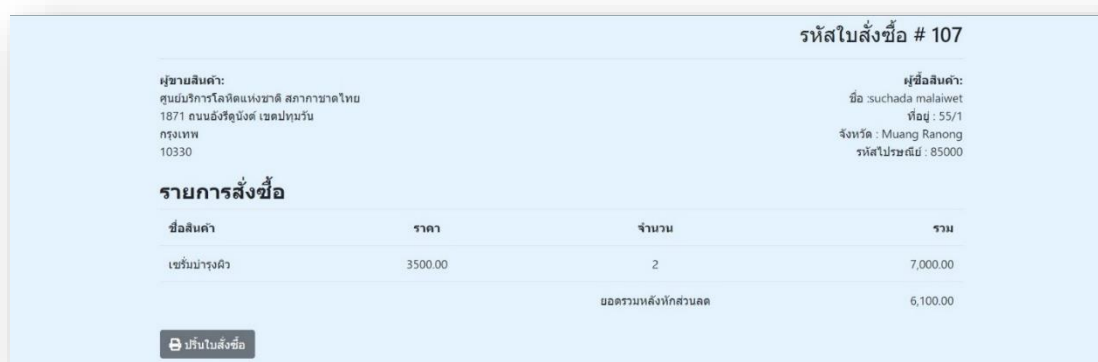


รูปที่ 5-16 แสดงหน้า Thank You! หลังทำการสั่งซื้อสำเร็จ

ผู้ใช้งานสามารถเข้าไปดูประวัติการสั่งซื้อได้และสามารถดูรายละเอียดการสั่งซื้อได้ด้วยเช่นกัน



รูปที่ 5-17 ประวัติการสั่งซื้อสินค้า



รูปที่ 5-18 รายละเอียดใบสั่งซื้อสินค้า

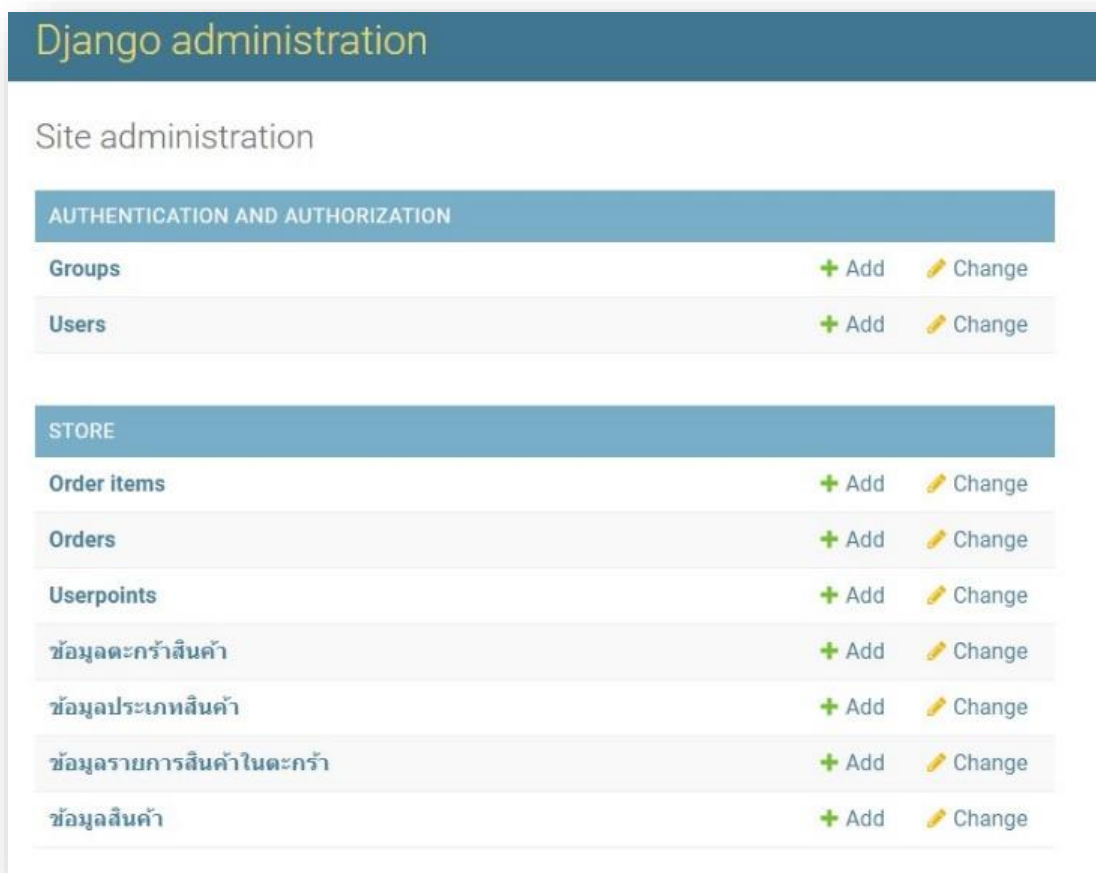
- Back-end ของ Admin

ในส่วนนี้ทาง Django framework ได้จัดทำไว้ให้ ขั้นแรกให้กรอกข้อมูล Super User(Admin) คือ UsernameและPassword ถ้ากรอกไม่ถูกต้องหรือไม่ตรงกับที่สร้างไว้ก็จะมีแจ้งเตือน

The image shows a Django administration login form. It has a blue header bar with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". The "Username:" field is a simple text box, and the "Password:" field is a text box with a light blue background. Below the password field is a blue button with the text "Log in". The entire form is enclosed in a light gray border with a subtle shadow.

รูปที่ 5-19 login ของ Admin

เมื่อผู้ดูแลระบบเข้าสู่ระบบแล้ว จะแสดงหน้าหลักที่แสดงรายการจัดการภายในร้านเช่น สินค้า, รายการสั่งซื้อของลูกค้า ผู้ดูแลระบบสามารถดูหรือจัดการกับรายการเหล่านี้ได้



รูปที่ 5-20 หน้า home ของ Admin

เมื่อผู้ดูแลระบบคลิกเข้าไปในปุ่ม User ก็จะมีข้อมูลของผู้ใช้ที่ได้มีการลงทะเบียนกับระบบไว้รวมถึงของผู้ดูแลระบบด้วย ที่เมื่อกดที่รายชื่อผู้ใช้นั้นๆจะสามารถดูข้อมูลเพิ่มเติมที่ผู้ใช้ลงทะเบียนเข้ามา รหัสผ่านของผู้ใช้จะถูกเข้ารหัสอีกครั้งที่ Django จัดการไว้ให้

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/> chuntika	chuntika.bum@gmail.com			<input checked="" type="checkbox"/>
<input type="checkbox"/> janejane	jane@gmail.co	jane	bumrung	<input type="checkbox"/>
<input type="checkbox"/> manamane	mana@gmail.com	mana	manee	<input type="checkbox"/>
<input type="checkbox"/> manee	manee1@gmail.com	manee	meena	<input type="checkbox"/>
<input type="checkbox"/> monday	monday@gmail.com	mon	bumrung	<input type="checkbox"/>
<input type="checkbox"/> ploy	ploy@gmail.com	suchada	malaiwet	<input type="checkbox"/>
<input type="checkbox"/> Tues	tues@gmail.com	Tuesday	Bumrung	<input type="checkbox"/>
<input type="checkbox"/> wednesday	wednesday@gmail.com	wednesday	bumrung	<input type="checkbox"/>

รูปที่ 5-21 แสดงผู้ใช้ที่ทำการลงทะเบียนไว้

Change user

Username: ploy
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: algorithm: pbkdf2_sha256 iterations: 216000 salt: 1MOU9e***** hash: xdB7QJ*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name: suchada

Last name: malaiwet

Email address: ploy@gmail.com

Permissions

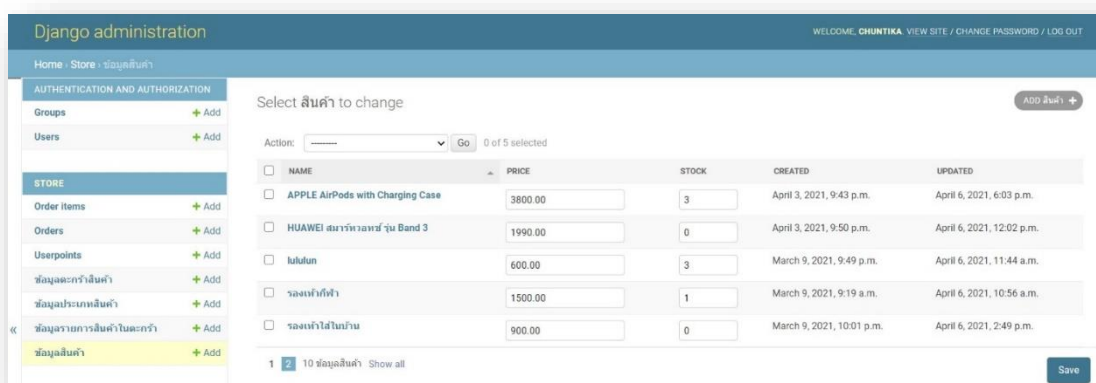
☒ **Active**
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ **Staff status**
Designates whether the user can log into this admin site.

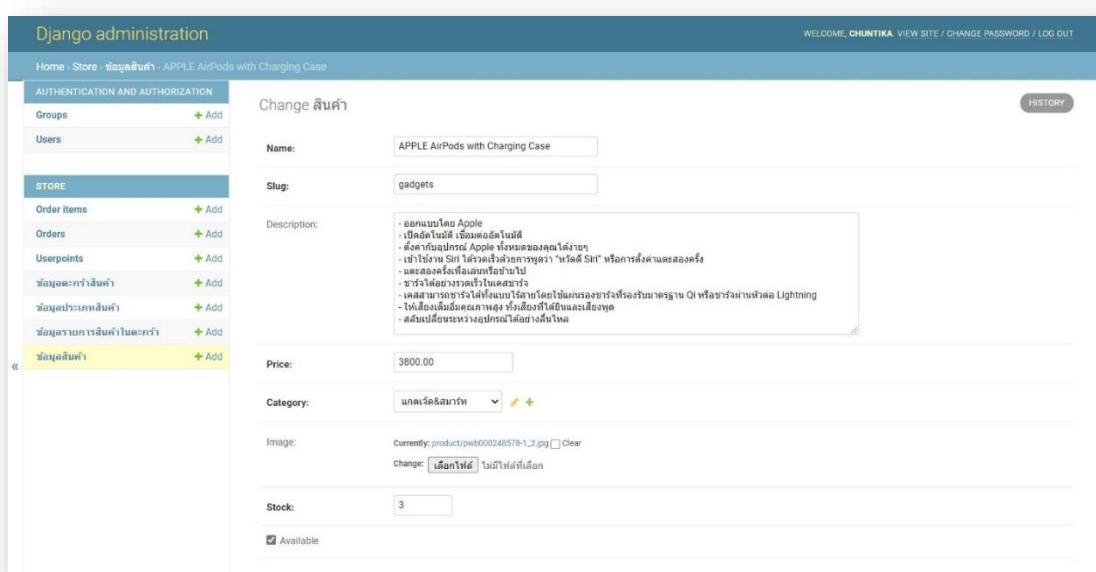
☐ **Superuser status**
Designates that this user has all permissions without explicitly assigning them.

รูปที่ 5-22 ข้อมูลของผู้ใช้

ถ้าผู้ดูแลระบบต้องการจัดการข้อมูลสินค้าในร้านค้า ให้กดที่ข้อมูลสินค้า เบื้องต้นหนึ่งแสดงนี้สามารถจัดการราคา, จำนวนสินค้าในคลัง แต่ถ้าอยากจัดการแบบละเอียดอย่าง คำอธิบายสินค้าหรือรูปภาพ ให้กดเข้าไปในชื่อของสินค้านั้นๆ หรือถ้าต้องการเพิ่มสินค้าเข้าร้านก็มีปุ่มกดเพิ่มสินค้าด้านขวาว่า “add สินค้า”

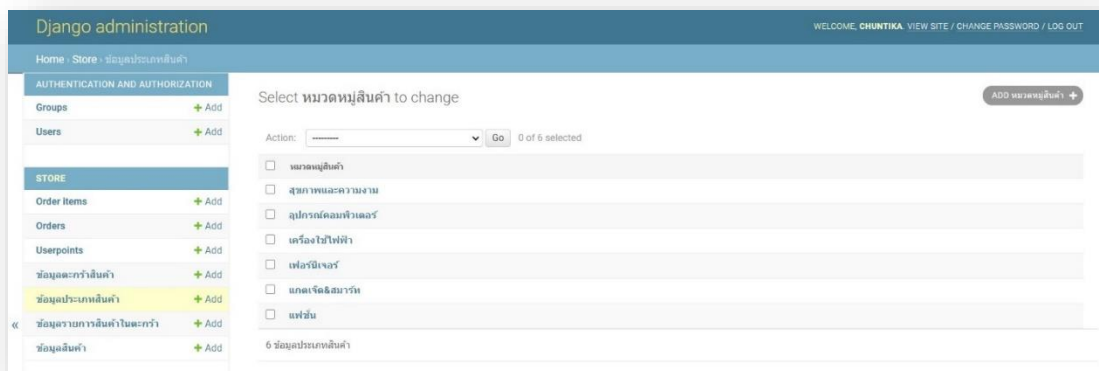


รูปที่ 5-23 หน้าข้อมูลสินค้า



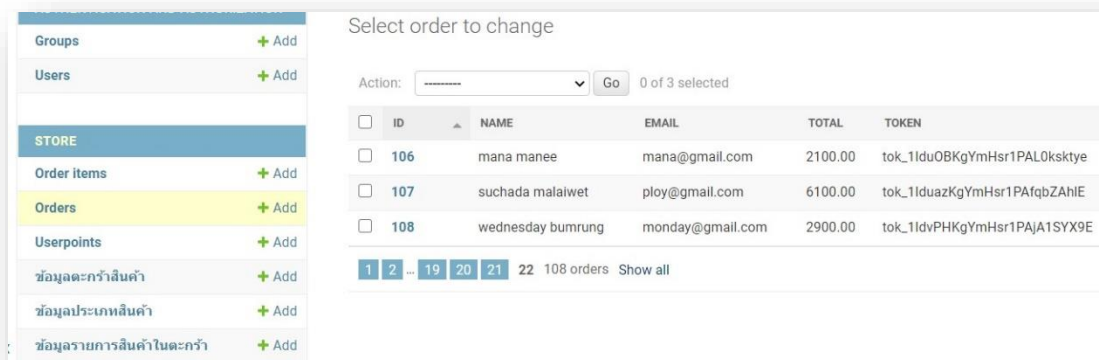
รูปที่ 5-24 ข้อมูลสินค้าแบบละเอียด

ถ้าต้องการจัดการของหมวดหมู่สินค้าสามารถกดเข้าไปที่ข้อมูลประเภทสินค้าก็จะจัดการเหมือนรูปแบบของข้อมูลสินค้าได้เช่นกัน



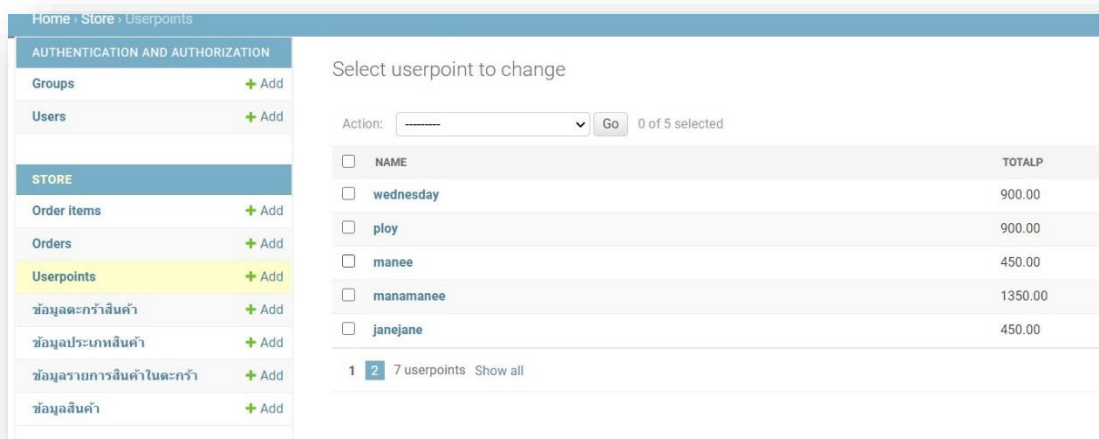
รูปที่ 5-25 หมวดหมู่ของสินค้าภายในร้านค้า

ในหน้า Orders จะมีคำสั่งซื้อของผู้ใช้ที่ทำการสั่งซื้อเข้ามาโดยคำสั่งซื้อจะเรียงตามรหัสคำสั่งซื้อ



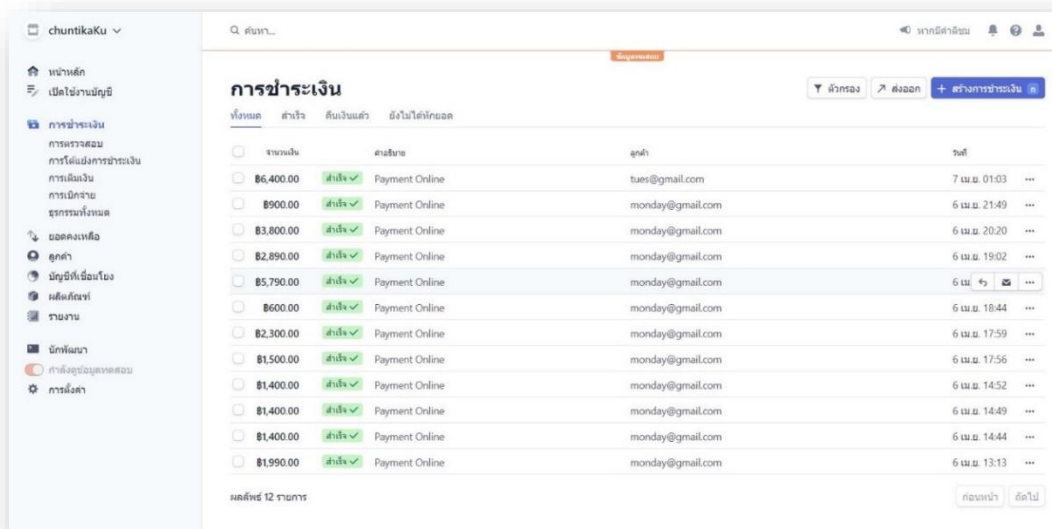
รูปที่ 5-26 คำสั่งซื้อภายในร้านค้า

แต่มีคะแนนของผู้ใช้เราได้ทำการจำลองขึ้นมา เพราะเราไม่สามารถเข้าถึงข้อมูลของสมาชิกขาดได้โดยตรง ในส่วนนี้ผู้ดูแลระบบจะเป็นผู้คอยอัปเดตแต้มคะแนนสะสมให้ผู้ใช้



รูปที่ 5-27 แต้มคะแนนสะสมของผู้ใช้

เมื่อผู้ดูแลระบบต้องการเช็คค่าผู้ใช้ได้ทำการชำระเงินผ่านทางระบบของเราสำเร็จหรือไม่ ผู้ดูแลระบบมีช่องทางที่สามารถดูประวัติการชำระเงินของผู้ใช้ได้ผ่านเซิร์ฟเวอร์ของ Stripe ที่ผู้ดูแลระบบได้ลงทะเบียนและเชื่อมกับระบบร้านไว้แล้ว โดยทาง Stripe จะแสดงให้ทั้งหมด



รูปที่ 5-28 Admin เช็คการชำระเงินผ่านทาง Stripe API

บทที่ 6

บทสรุป และข้อเสนอแนะ

บทนี้จะกล่าวถึงบทสรุปในการพัฒนาโครงการ ปัญหาและอุปสรรค วิธีแนวทางการแก้ไข แนวทางการพัฒนาโครงการ รวมถึงข้อเสนอแนะ

6.1. สรุปผล

ในการพัฒนาโครงการ HiBLOOD Shop Web Application ได้เป็นไปตามขั้นตอนที่ออกแบบไว้เกือบ 100% แต่มีความผิดพลาดระหว่างพัฒนา เนื่องจากยังขาดส่วนของการติดตามสินค้าทำได้เพียงโทรหาทางสถานกาชาด จึงเปลี่ยนตามความเหมาะสม ให้ใกล้เคียงกับที่วางแผนดำเนินงานไว้ โครงการสามารถนำไปใช้งานได้จริงและสามารถต่อยอดเพื่อวัตถุประสงค์อื่นๆได้อีกในอนาคต

6.2. ปัญหาและอุปสรรค

ในการพัฒนา HiBLOODShopWebApplication มีอุปสรรคในการดำเนินงานอย่างแรก เนื่องจากในระยะเวลาที่น้อย ทำให้ต้องปรับเปลี่ยนโครงการไม่เหมือนที่ได้เสนอกับอาจารย์ในครั้งแรก และการใช้โปรแกรมที่ไม่มีความชำนาญมากพอ และอุปสรรคใหญ่คือ สถานการณ์ Covid-19 ทำให้ต้องดำเนินการติดต่อสื่อสารผ่านทางช่องทางออนไลน์เท่านั้นและผู้จัดทำร่วมกันได้ประสบอุบัติเหตุทำให้การเดินทางเพื่อพัฒนาโครงการร่วมกันไม่สะดวกแต่โดยรวมแล้วโครงการค่อนข้างสมบูรณ์สามารถใช้งานได้จริง

6.3.แนวทางการแก้ไข

ทางคณะผู้จัดทำจึงปรึกษากันเพื่อปรับเปลี่ยนโครงการเพื่อให้ทันกับระยะเวลาที่กำหนดไว้และให้งานออกมาสมบูรณ์ ในด้านการใช้โปรแกรมได้ศึกษาหาข้อมูลจากหลายช่องทาง เช่น หนังสือ, อินเทอร์เน็ตหรือ youtube เพื่อทำความเข้าใจเพิ่มมากขึ้นและหาวิธีแก้ไขเมื่อโปรแกรมมีปัญหา

6.4. แนวทางพัฒนา

- การวิเคราะห์ปัญหา

ได้ลองบริจาคเลือดเพื่อดูการทำงานของศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย ดูจำนวนคนที่ไปบริจาคเลือดว่ามีเยอะหรือไม่และแบ่งหน้าที่ให้แต่ละคนตามความเหมาะสมและความถนัดและพบปัญหาของการบริจาคเลือดที่ว่าจำนวนผู้เข้ารับการบริจาคเลือดลดลงทำให้เกิดการขาดแคลนของเลือดเป็นอย่างมาก

- การออกแบบระบบ

เน้นออกแบบให้ผู้ใช้งานง่ายและใช้งานได้จริง ไม่ซับซ้อน ทุกคนที่ใช้สามารถเข้าใจและเข้าถึงได้และใช้ ประโยชน์จากการบริจาคเลือดเพื่อให้เกิดเป้าหมายอื่นๆเพิ่มขึ้น

6.5. ข้อจำกัดของระบบ

เว็บจะตัดแต้มคะแนนทั้งหมดเท่านั้น ไม่สามารถเลือกใช้จำนวนแต้มคะแนนได้

6.6. ข้อเสนอแนะ

- เป้าหมาย: เพื่อต่อยอด และนำไปสู่การใช้ได้จริง
- เวลา: เวลาในการทำโครงการน้อยเกินไป
- ต้นทุน: ไม่มีเพราะทุกโปรแกรมที่ใช้เป็น open source จึงไม่มีค่าใช้จ่าย
- คุณภาพ: หลังจากทดสอบระบบการทำงานแล้ว ระบบทำงานได้ถูกต้องและครบถ้วนสมบูรณ์
- ประสิทธิภาพ: ระบบทำงานมีความถูกต้องเกือบ 100%

- ตอบคำถาม

1. ลูกค้าจะสามารถทราบได้อย่างไรว่าสินค้าได้รับการจัดส่งแล้วหรือมีสถานะเป็นอย่างไร

: ทาง admin จะแจ้งเลข track พัสดุผ่านทาง email เพราะเราเก็บข้อมูล email มาจากการลงทะเบียนของ user

2. ลูกค้าจะสามารถติดต่อกลับได้อย่างไรเมื่อสินค้าที่ส่งมีปัญหาหรือเกิดการชำรุดระหว่างทาง

: ศูนย์บริการโลหิต สภากาชาด มีเบอร์ติดต่อ ลูกค้าสามารถโทรมาเมื่อสินค้ามีปัญหาได้ แต่เรื่องการชำรุดระหว่างทางนั้นจะอยู่ในความรับผิดชอบของทางบริษัทขนส่ง

3. การเพิ่มแต้มให้กับลูกค้าจำเป็นจะต้องให้ผู้ดูแลระบบเข้ามาเพิ่มในระบบด้วยตัวเองอย่างเดียวใช่ไหม

: admin จัดการเท่านั้น เพื่อกันกรณีไม่ได้บริจาคจริง

4. ออกแบบระบบได้ครบถ้วนตามที่ระบบ e-commerce ควรจะมีครับ แล้วก็รู้สึกสนใจใน stripe API ว่าถ้าต้องการใช้งานกับเว็บไซต์ของเราบ้างจะต้องมีขั้นตอนการ implement อย่างไรบ้างครับ:

- ไปที่ <https://stripe.com/th-us> จากนั้น กดปุ่ม start now เพื่อทำการสร้าง account ขึ้นมา

- เมื่อลงทะเบียน create account เรียบร้อย, activate และ verify ผ่าน email เพื่อให้ account สมบูรณ์ กดปุ่ม Add name ก็จะได้บัญชีสำหรับ stripe แล้วสำหรับสำหรับการดำเนินการชำระเงินที่วิ่งเข้าระบบเรา

- กลับมาที่ตัว project ของเรา, ติดตั้ง library stripe ด้วยคำสั่ง pip install stripe

- ไปที่ไฟล์ setting.py (project พัฒนาด้วย python | Django framework) ในส่วน INSTALLED_APPS ใส่คำสั่ง 'stripe' โลบราลีจากภายนอกจะสามารถมองหา stripe เจอแล้ว

- สร้างตัวแปรที่เก็บ API key ที่เป็นรูปแบบ static คือ PUBLIC_KEY กับ SECRET_KEY เพื่ออ้างอิงและทำงานร่วมกับโปรเจค โดยไปที่ที่หน้าเว็บ stripe เลือก Get your test API keys แล้วทำการคัดลอก public key กับ secret key มาใส่ในตัวแปรที่เราสร้าง

- เรียบร้อยสำหรับการนำ API มาใช้กับตัวโปรเจค

บรรณานุกรม

1. ทำไมต้องบริจาคโลหิต?. (2563). [ออนไลน์]. ได้จาก:
<https://www.redcross.or.th/news/information/13117/> [สืบค้นเมื่อวันที่ 29 มีนาคม 2564]
2. อูสา สุทธิสาคร, สิทธิพงษ์พัฒนานนท์สกุล, ดำรง เชี่ยวศิลป์. (2557). พฤติกรรมการบริจาคโลหิตของผู้ที่มาบริจาคโลหิต ณ ศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย.วารสารโลหิตวิทยาและเวชศาสตร์บริการโลหิต. [ออนไลน์]. 24(3), 251-252. ได้จาก: [http://www.tsh.or.th/file_upload/files/05\(2\).pdf](http://www.tsh.or.th/file_upload/files/05(2).pdf) [สืบค้นเมื่อวันที่ 29 มีนาคม 2564]
3. สายทอง วงศ์คำ. (2558). ปัจจัยที่มีผลต่อการบริจาคเลือดของผู้ที่มาบริจาคเลือด ณ ธนาคารเลือด โรงพยาบาลหัวหิน. วารสารวิทยบริการ มหาวิทยาลัยสงขลานครินทร์. [ออนไลน์]. 26(2), 83. ได้จาก:
<https://journal.oas.psu.ac.th/index.php/asj/article/viewFile/790/824> [สืบค้นเมื่อวันที่ 29 มีนาคม 2564]
4. Wasin Thiengkunakrit. (2560). เริ่มพัฒนา Web Application กับภาษา Python ด้วย Django Framework. [ออนไลน์]. ได้จาก:
<https://codeburst.io/%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%9E%E0%B8%B1%E0%B8%92%E0%B8%99%E0%B8%B2-web-application-%E0%B8%81%E0%B8%B1%E0%B8%9A%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2-python-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-django-framework-38ce132ac706> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]

5. Smith Krengkrud. (2560). คุณรู้จัก Django หรือเปล่า (ดีจังอ่าว). [ออนไลน์]. ได้จาก:
<https://medium.com/@mitjy/%E0%B8%84%E0%B8%B8%E0%B8%93%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-django-%E0%B8%AB%E0%B8%A3%E0%B8%B7%E0%B8%AD%E0%B9%80%E0%B8%9B%E0%B8%A5%E0%B9%88%E0%B8%B2-%E0%B8%94%E0%B8%B5%E0%B8%88%E0%B8%B1%E0%B8%87%E0%B8%87%E0%B9%88%E0%B8%AD%E0%B8%A7%E0%B8%A7-72fcb4d11aa3> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]
6. ดร.ณัฐพล แสนคำ. (2563). วิธีการใช้งาน Visual Studio Code. [ออนไลน์]. ได้จาก:
<http://cs.bru.ac.th/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89-visual-studio-code-2/> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]
7. การติดตั้ง XAMPP. (2560). [ออนไลน์]. ได้จาก:
<http://www.ruyoung.com/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%95%E0%B8%B4%E0%B8%94%E0%B8%95%E0%B8%B1%E0%B9%89%E0%B8%87-xampp/>
 [สืบค้นเมื่อวันที่ 13 เมษายน 2564]
8. phpMyAdmin คืออะไร มีวิธีการใช้งานอย่างไร. (2563). [ออนไลน์]. ได้จาก:
<https://www.webdodee.com/what-is-phpmyadmin/> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]

9. Developer. (2563). Bootstrap คืออะไร? ใช้งานยังไง?. [ออนไลน์]. ได้จาก: <https://www.borntodev.com/c/webdeveloper/bootstrap-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87-5fead4a368fff> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]
10. Twometre. (2558). จับตา Stripe สตาร์ทอัพน้องใหม่ เปลี่ยนธุรกรรมออนไลน์ให้ง่ายเหมือนขนม. [ออนไลน์]. ได้จาก: <https://www.blognone.com/node/65401> [สืบค้นเมื่อวันที่ 13 เมษายน 2564]
11. Natnalin Sompai boon. (2561). การใช้งาน Figma ออกแบบ UI (Basic). [ออนไลน์]. ได้จาก: <https://medium.com/teamappman/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-figma-%E0%B8%AD%E0%B8%AD%E0%B8%81%E0%B9%81%E0%B8%9A%E0%B8%9A-ui-basic-6675da0a0926> [สืบค้นเมื่อวันที่ 19 เมษายน 2564]

12. Darkcrist. DBeaver. เครื่องมือที่ยอดเยี่ยมสำหรับการจัดการฐานข้อมูลต่างๆ. [ออนไลน์]. ได้จาก:

<https://blog.desdelinux.net/th/dbeaver->

%E0%B9%80%E0%B8%9B%E0%B9%87%E0%B8%99%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%A1%E0%B8%B7%E0%B8%AD%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%A2%E0%B8%AD%E0%B8%94%E0%B9%80%E0%B8%A2%E0%B8%B5%E0%B9%88%E0%B8%A2%E0%B8%A1%E0%B8%AA%E0%B8%B3%E0%B8%AB%E0%B8%A3%E0%B8%B1%E0%B8%9A%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%88%E0%B8%B1%E0%B8%94%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%86/#comments [สืบค้นเมื่อ

วันที่ 19 เมษายน 2564]

ภาคผนวก

- ประวัติผู้จัดทำโครงการ



ชื่อ	สุชาดา มาลัยเวช
ชื่อเล่น	พลอย
วันเกิด	2 ธันวาคม 2541
สาขาวิชา	วิทยาการคอมพิวเตอร์
สถานศึกษา	มหาวิทยาลัยเกษตรศาสตร์
ที่อยู่ปัจจุบัน	46/107 ม.สีรินเฮาส์ ซ.ประชาอุทิศ 18 ถ. ช่างอากาศอุทิศ แขวงดอนเมือง เขตดอน เมือง กรุงเทพฯ 10210

รูปที่ 7-4 ผู้จัดทำ สุชาดา มาลัยเวช



ชื่อ	ฉันทิกา บำรุง
ชื่อเล่น	นิว
วันเกิด	10 กรกฎาคม 2541
สาขาวิชา	วิทยาการคอมพิวเตอร์
สถานศึกษา	มหาวิทยาลัยเกษตรศาสตร์
ที่อยู่ปัจจุบัน	ริณวรารพาร์ทเมนต์ 55/1 5B2 ซอย พหลโยธิน34 ถนนพหลโยธิน แขวงเสนา นิคม เขตจตุจักร กรุงเทพฯ 10900

รูปที่ 7-5 ผู้จัดทำ ฉันทิกา บำรุง